```
B.JACKIE

DEST=B.JACKIE  USER=B.JACKIE  QUEUE=LPT  DEVICE=@P1
SEQ=40089  QPRI=127  LPP=66  CPL=132  COPIES=1  LIMIT=767

              CREATED:    7-JUN-83  15:02:44
             ENQUEUED:    7-JUN-83  15:50:54
             PRINTING:    7-JUN-83  15:50:55

              PATH=:UDD:B.JACKIE:PILOT.PRN

PILOT.PRN

Atari S/W Development HCD1

AOS/VS REV 02.08
AOS/VS XLPT REV 01.61
```

```
+44444444444444444444444444444444444444444444444444444444444444444444444444444444444444444444444444444444444444444444444444444444+

                    $$$$           $    $     $$$   $   $  $$$$$  $$$$$
                    $   $          $   $ $   $   $  $  $     $    $
                    $   $          $  $   $  $      $ $      $    $
                    $$$$           $  $   $  $      $$       $    $$$$
                    $   $          $  $$$$$  $      $ $      $    $
                    $   $  $$  $   $  $   $  $   $  $  $     $    $
                    $$$$   $$   $$$   $   $   $$$   $   $  $$$$$  $$$$$

                     DEST=B.JACKIE  USER=B.JACKIE  QUEUE=LPT  DEVICE=@p1
                    SEQ=44089  QPRI=127  LPP=66  CPL=132  COPIES=1  LIMIT=767

                                 CREATED:    7-JUN-83  15:02:44
                                ENQUEUED:    7-JUN-83  15:50:54
                                PRINTING:    7-JUN-83  15:50:55

                                 PATH=:UDD:B.JACKIE:PILOT.PRN

                 $$$$   $$$$$  $       $$$   $$$$$      $$$$   $$$$   $   $
                 $   $    $    $      $   $    $        $   $  $   $  $$  $
                 $   $    $    $      $   $    $        $   $  $   $  $ $ $
                 $$$$     $    $      $   $    $        $$$$   $$$$   $  $$
                 $        $    $      $   $    $        $      $ $    $   $
                 $        $    $      $   $    $    $$  $      $  $   $   $
                 $      $$$$$  $$$$$   $$$     $    $$  $      $   $  $   $

+44444444444444444444444444444444444444444444444444444444444444444444444444444444444444444444444444444444444444444444444444444444+


                                Atari S/W Development HCD1


                                   AOS/VS REV 02.08
                                 AOS/VS XLPT REV 01.61
```

PILOT46.OBJ

```
                        ;
                        ; PILOT PROGRAM EQUATE FILE
                        ;
                        ; EDIT #99 -- 07-JUN-83
                        ;

      = 0000            DEBUG   = 0                 ; INCLUDE DEBUG CODE IF 1, DON'T IF 0.
      = 0000            LITPEN  = 0                 ; INCLUDE LIGHTPEN CODE IF 1, DON'T IF 0.
      = 0000            LOGGRP  = 0                 ; INCLUDE LOGICAL OPERATORS IF 1, DON'T IF 0.
      = 0000            DOS     = 0                 ; INCLUDE 'DOS' COMMAND IF 1, DON'T IF 0.
      = 0000            FALSE   = 0

                        ; COLLEEN SYSTEM I/O

      = E456            CIO     = $E456

      = E400            IOVBAS  = $E400             ; COLLEEN VECTOR BASE ADDRESS.
      = E406            EPUTC   = $E406             ; "E:" PUT CHARACTER.
      = E414            SGETC   = $E414             ; "S:" GET CHARACTER.
      = E416            SPUTC   = $E416             ; "S:" PUT CHARACTER.
      = E41A            SSPEC   = $E41A             ; "S:" SPECIAL.
      = E424            KGETC   = $E424             ; "E:" GET CHARACTER.

      = 0010            IOCBSZ  = 16                ; # OF BYTES PER IOCB.
      = 0000            IOCB0   = $00               ; CONSOLE INPUT/OUTPUT.
      = 0010            IOCB1   = IOCB0+IOCBSZ      ; (UNUSED).
      = 0020            IOCB2   = IOCB1+IOCBSZ      ; GRAPHICS INPUT & OUTPUT.
      = 0030            IOCB3   = IOCB2+IOCBSZ      ; LOAD & SAVE I/O.
      = 0040            IOCB4   = IOCB3+IOCBSZ      ; IN: & OUT: USE IOCB4 THRU IOCB7.
      = 0070            IOCB7   = 3*IOCBSZ+IOCB4

      = 0340            ICHID   = $0340             ; IOCB HANDLER I.D.
      = 0341            ICDNO   = ICHID+1           ; DEVICE #.
      = 0342            ICCOM   = ICDNO+1           ; COMMAND BYTE.
      = 0343            ICSTA   = ICCOM+1           ; STATUS BYTE.
      = 0344            ICBAL   = ICSTA+1           ; BUFFER ADDRESS (LO).
      = 0345            ICBAH   = ICBAL+1           ; BUFFER ADDRESS (HI).
      = 0346            ICPLL   = ICBAH+1           ; RECORD LENGTH (LO).
      = 0347            ICRLH   = ICRLL+1           ; RECORD LENGTH (HI).
      = 0348            ICBLL   = ICRLH+1           ; BUFFER LENGTH (LO).
      = 0349            ICBLH   = ICBLL+1           ; BUFFER LENGTH (HI).
      = 034A            ICAUX1  = ICBLH+1           ; AUX1.
      = 034B            ICAUX2  = ICAUX1+1          ; AUX2.

      = 0022            ICCOMZ  = $0022             ; ZERO PAGE IOCB COMMAND BYTE.

      = 0003            OPEN    = $03               ; OPEN COMMAND.
      = 000C            CLOSE   = $0C               ; CLOSE COMMAND.
      = 0007            GETC    = $07               ; GET CHARACTER COMMAND.
      = 000B            PUTC    = $0B               ; PUT CHARACTER COMMAND.
      = 0005            GETR    = $05               ; GET RECORD COMMAND.
      = 0009            PUTR    = $09               ; PUT RECORD COMMAND.

      = 0004            OREAD   = $04               ; OPEN DIRECTION.
      = 0008            OWRIT   = $08               ; OPEN DIRECTION.
      = 0010            SPLIT   = $10               ; SPLIT SCREEN OPTION.
      = 0020            NOCLR   = $20               ; INHIBIT SCREEN CLEAR OPTION.
```

```
   = 000D       CR       = $0D             ; INTERNAL <CR> CODE

   = 001C       CUP      = $1C             ; CURSOR UP.
   = 001D       CDOWN    = $1D             ; CURSOR DOWN.
   = 001E       CLEFT    = $1E             ; CURSOR LEFT.
   = 001F       CRIGHT   = $1F             ; CURSOR RIGHT.
   = 009B       EOL      = $9B             ; ATASCII END OF LINE.
   = 007D       CLEAR    = $7D             ; MONITOR CLEAR SCREEN.
   = 00FD       BELL     = $FD             ; BELL CODE.
   = 005C       BSLASH   = $5C             ; BACKSL SLASH
   = 007C       VBAR     = $7C             ; VERTICAL BAR
   = 005B       SBRACK   = $5B             ; SQUARE BRACKET
   = 0027       SQUOTE   = $27             ; SINGLE QUOTE
   = 007F       TAB      = $7F             ; TAB.

                ; O.S. ROM VECTORS

   = E462       XITVBV   = $E462           ; EXIT VBLANK VECTOR.
   = E45C       SETVBV   = $E45C           ; SET VECTOR ROUTINE.

                ; O.S. DATA BASE

   = 0244       COLDST   = $0244           ; SYSTEM COLDSTART FLAG.
   = 000C       DOSINI   = $000C
   = 006A       RAMTOP   = $006A           ; TOP OF SCREEN ADDRESS (MSB).
   = 004F       COLRSH   = $004F           ; ATTRACT HUE SHIFT.
   = 004E       DRKMSK   = $004E           ; ATTRACT LUM LIMIT.
   = 02E7       MEMLO    = $02E7           ; LOWEST AVAILABLE RAM [WORD].
   = 02E5       MEMHI    = $02E5           ; HIGHEST AVAILABLE RAM [WORD].
   = 000E       APPMHI   = $000E           ; APPLICATION MEM HI [WORD].
   = 0011       BREAK    = $0011           ; BREAK KEY FLAG.
   = 02FC       CH       = $02FC           ; KEYBOARD MATRIX CODE INPUT.
   = 02F0       CRSINH   = $02F0           ; CURSOR INHIBIT FLAG.
   = 02FE       DSPFLG   = $02FE           ; CONTROL BYTE DISPLAY FLAG.
   = 0230       SDLSTL   = $0230           ; DISPLAY LIST ADDRESS.
   = 0232       SSKCTL   = $0232
   = 0012       RTCLOK   = $0012           ; 60 HZ CLOCK.
   = 0052       LMARGN   = $0052           ; SCREEN LEFT MARGIN.
   = 0053       RMARGN   = $0053           ; SCREEN RIGHT MARGIN.
   = 0055       COLCRS   = $0055           ; SCREEN COLUMN [WORD].
   = 0054       ROWCRS   = $0054           ; SCREEN ROW [BYTE].
   = 0057       DINDEX   = $0057           ; S: SCREEN MODE.
   = 0058       SAVMSC   = $0058           ; SCREEN START ADDR.
   = 02BF       BOTSCR   = $02BF           ; TEXT SCREEN SIZE.
   = 0008       WARMST   = $0008           ; WARMSTART FLAG (0 IF POWERUP).
   = 000A       DOSVEC   = $000A           ; DOS START VECTOR.
   = 026E       FINE     = $026E           ; SCROLL SELECT.
   = 026F       GPRIOR   = $026F           ; PLAYER/PLAYFIELD PRIORITY.
   = 022F       DMACT    = $022F           ; DMA CONTROL BYTE.
   = 02C0       PCOLR0   = $02C0           ; PLAYER/MISSILE COLOR.
   = 02C4       COLOR0   = $02C4           ; COLOR REGISTER 0 VALUE.
   = 0270       PADDL0   = $0270           ; PADDLE CONTROLLER 0.
   = 0278       STICK0   = $0278           ; JOYSTICK 0.
   = 027C       PTRIG0   = $027C           ; PADDLE TRIGGER 0.
   = 0284       STRIG0   = $0284           ; JOYSTICK TRIGGER 0.
   = 0234       LPENH    = $0234           ; LIGHTPEN HORIZONTAL POSITION.
```

```
      = 0235      LPENV    = $0235          ; LIGHTPEN VERTICAL POSITION.
      = 0291      TXTCOL   = $0291          ; SPLIT SCREEN TEXT COLUMN.
      = 0290      TXTROW   = $0290          ; SPLIT SCREEN TEXT ROW.
      = 02B6      INVFLG   = $02B6          ; INVERSE VIDEO FLAG FOR KEYBOARD.
      = 0200      VDSLST   = $0200          ; DISPLAY LIST INTERRRUPT.
      = 0220      CDTMV5   = $0220          ; SYSTEM TIMER VALUE.
      = 0224      VVBLKD   = $0224          ; DEFERRED VBLANK ROUTINE.
      = 02FF      SSFLAG   = $02FF          ; START/STOP FLAG.

                  ; PILOT ERROR CODES

      = 0080      NS       = $80            ; "NOT A SYNTAX ERROR" FLAG.
      = 0001      RDYTXT   = 1              ; READY.
      = 0081      EOPERR   = 1+NS           ; END OF PROGRAM STORAGE REACHED DURING RUN.
      = 0081      AUTUXT   = 1+NS           ; EXIT AUTO-INPUT MODE.
      = 0002      CNDERR   = 2              ; CONDITION FIELD ERROR (':' EXPECTED).
      = 0002      NSTERR   = 2              ; GRAPHICS SUB-COMMAND NESTING ERROR.
      = 0081      ENDERR   = 1+NS           ; USE STACK EMPTY ON END COMMAND.
      = 0002      JNKERR   = 2              ; JUNK AT END OF STATEMENT.
      = 0086      IOERR    = 6+NS           ; I/O ERROR.
      = 0002      IVCERR   = 2              ; INVALID COMMAND.
      = 0002      ATMERR   = 2              ; INVALID ATOM SYNTAX.
      = 0002      IMPERR   = 2              ; IMPROPER COMMAND PARAMETER.
      = 0089      INSERR   = 9+NS           ; INSUFFICIENT STORAGE FOR OPERATION.
      = 0087      ABTERR   = 7+NS           ; OPERATOR ABORT.
      = 000A      UNDERR   = 10             ; UNDEFINED LABEL OPERAND.
      = 008B      USOERR   = 11+NS          ; USE STACK OVERFLOW.
      = 0002      EXPERR   = 2              ; EXPRESSION ERROR.
      = 008C      INTERR   = 12+NS          ; INTERNAL BUG ERROR.
      = 008D      LNOERR   = 13+NS          ; LINE # OUT OF RANGE.
                  ;          14               IS RESERVED.
      = 000F      OLLERR   = 15             ; OVERLENGTH INPUT LINE.
                  ;          16-21            ARE RESERVED.
      = 0096      FILERR   = 22+NS          ; TOO MANY IN/OUTS.
      = 0017      SIGNON   = 23             ; POWER-UP SIGN-ON MESSAGE.
      = 0018      TRCMES   = 24             ; TRACE PREAMBLE.
      = 0008      ASTMES   = 8              ; ASTERISKS.
      = 0083      NRCERR   = 3+NS           ; NOT CORRECT COMMAND MODE.
      = 0084      DIVERR   = 4+NS           ; DIVIDE BY ZERO.
      = 0085      SCNERR   = 5+NS           ; SCREEN MODE CONFLICT.
      = 0099      CNTERR   = 25+NS          ; CAN'T CONTINUE.
      = 009A      STPMES   = 26+NS          ; STOP.
      = 009B      RENERR   = 27+NS          ; CAN'T RENUMBER
      = 009C      OVLPER   = 28+NS          ; OVERLAPPING RANGE.
      = 009D      TOMES    = 29+NS          ; " TO ".
      = 009E      NCHGMS   = 30+NS          ; PROGRAM IS UNCHANGED.
      = 009F      DELMES   = 31+NS          ; "YOU ARE ABOUT TO DELETE ".
      = 00A0      DL2MES   = DELMES+1       ; LINES.<CR>ARE YOU SURE?".
      = 0021      SPTERR   = 33             ; NO SPLIT SCREEN.
      = 0022      MODERR   = 34             ; INVALID GRAPHICS MODE.
      = 00A4      FSOFER   = 36+NS          ; FLOOD STACK OVERFLOW.
      = 0025      NMCERR   = 37             ; NO MORE COLORS.
      = 0026      DCAERR   = 38             ; DOUBLE COLOR ASSIGN.

                  ; ATOM IDENTIFIER CODES (PRODUCED BY 'ATOM')

      = 0001      NULL     = 1              ; NULL ATOM.
      = 0001      NULL    = 1               ; NULL ATOM.
```

```
      = 0002      NUM     = 2               ; NUMERIC CONSTANT.
      = 0004      NVAR    = 4               ; NUMERIC VARIABLE OR POINTER TO WORD.
      = 0008      SVAR    = 8               ; STRING VARIABLE.
      = 0010      USVAR   = 16              ; UNDEFINED STRING VARIABLE.
      = 0020      TEXT    = 32              ; TEXT.
      = 0040      OPR     = 64              ; OPERATOR.
      = 0080      BPTR    = 128             ; POINTER TO BYTE.

                  ; GRAPHICS OPERATORS

      = 0011      FILL    = $11
      = 0012      FILLTO  = $12
      = 0009      DRAW    = $09
      = 000A      DRAWTO  = $0A
      = 0005      GO      = $05
      = 0006      GOTO    = $06

                  ; EDGE DETECT STATUS BITS

      = 0008      ELEFT   = 8               ; LEFT EDGE.
      = 0004      ERIGHT  = 4               ; RIGHT EDGE.
      = 0002      EBOTOM  = 2               ; BOTTOM EDGE.
      = 0001      ETOP    = 1               ; TOP EDGE.

                  ; PILOT CONFIGURATION PARAMETERS.

      = 0030      USTKSZ  = 48              ; 24 LEVELS IN USE STACK.
      = 0002      ELEVEL  = 2               ; # OF EXPRESSION STACK () LEVELS.
      = 000E      ESTKSZ  = 4*ELEVEL+6      ; EXPRESSION STACK SIZE.
      = 00FE      ACCLNG  = 254             ; ACCEPT BUFFER LENGTH.
      = 00FE      TEXLNG  = 254             ; TEXT EXPRESSION BUFFER LENGTH.
      = 007A      LINLNG  = 122             ; COMMAND/ACCEPT INPUT LINE LENGTH.
      = 270F      MAXLN   = 9999            ; MAXIMUM PROGRAM LINE NUMBER.
      = 0004      AUREGS  = 4               ; 4 AUDIO REGISTERS.
      = 0007      SCNMOD  = 7               ; 4 COLOR, 160 * 96.
      = 000F      DNSIZE  = 15              ; DEVICE/FILENAME MAXIMUM LENGTH.
      = 0028      TCOL    = 40              ; TEXT SCREEN # OF COLUMNS.
      = 0018      TROW    = 24              ; TEXT SCREEN # OF ROWS.
      = 000A      INBFSZ  = 10              ; MAXIMUM SUBCOMMAND LENGTH.

                  ; HARDWARE EQUATES

      = D200      AUDF1   = $D200           ; AUDIO #1 FREQUENCY DIVIDER.
      = D201      AUDC1   = AUDF1+1         ; AUDIO #1 TYPE/VOLUME.
      = D20F      SKCTL   = $D20F           ; SERIAL PORT CONTROL.
      = D20F      SKSTAT  = $D20F           ; SERIAL PORT STATUS.
      = D208      AUDCTL  = $D208           ; AUDIO CONTROL REGISTER.
      = D302      PACTL   = $D302           ; PIA CASSETTE CONTROL.
      = 0034      CASSON  = $34             ; CASSETTE ON.
      = 003C      CASSOF  = $3C             ; CASSETTE OFF.
      = D20A      PKYRND  = $D20A           ; POKEY RANDOM NUMBER.
      = D01F      CONSOL  = $D01F           ; START/SELECT/OPTION KEY READ.
      = D400      DMACTL  = $D400           ; DMA CONTROL REG.
      = D407      PMBASE  = $D407           ; PLAYER/MISSILE BASE ADDRESS REGISTER.
      = D010      GRAFP3  = $D010           ; PLAYER 3 DATA.
      = D01D      GRACTL  = $D01D           ; GRAPHICS CONTROL REG.
      = D00B      SIZEP3  = $D00B           ; PLAYER 3 SIZE.
```

```
     = D000      HPOS0    = $D000          ; PLAYER POSITIONS.
     = D017      COLPF1   = $D017          ; PLAYFIELD 1 COLOR.
     = D018      COLPF2   = $D018          ; PLAYFIELD 2 COLOR.
     = D40A      WSYNC    = $D40A          ; WAIT FOR SYNC.
     = D40E      NMIEN    = $D40E          ; NMI ENABLE.

                 ; COLOR EQUATES

     = 0042      CRED     = $42
     = 0084      CBLUE    = $84
     = 001A      CYELLO   = $1A
     = 0001      CBLACK   = $01

                 ; MISCELLANEOUS

     = 0080      PCUP     = 128            ; PEN = 'UP'.
     = 0040      PCDN     = 64             ; PEN = 'DOWN'.

     = 0000      LSMLL    = 0              ; LETTERS = 'SMALL'.
     = 0001      LMED     = 1              ; LETTERS = 'MEDIUM'.
     = 0002      LLRG     = 2              ; LETTERS = 'LARGE'.

     = 0001      EWRAP    = 1              ; EDGE = 'WRAP'.
     = 0002      EHALT    = 2              ; EDGE = 'HALT'.
     = 0004      EBNC     = 4              ; EDGE = 'BOUNCE'.
     = 0008      EFREE    = 8              ; EDGE = 'FREE'.

                 ; ALGORITHMS REQUIRE KOFF=0, KON=1.

     = 0000      KOFF     = 0              ; 'OFF'.
     = 0001      KON      = 1              ; 'ON'.

     = FFFF      EONMLS   = $FFFF          ; END OF 'NMSBUF' LIST.

     = 00DF      UC       = $DF            ; LOWER -> UPPER CASE.
     = 0020      LC       = $20            ; UPPER -> LOWER CASE.

     = 00FF      UP       = $FF            ; FLOOD DIRECTIONS.
     = 0001      DOWN     = 1

     = 0001      STRTKY   = 1              ; CONSOLE KEY DEFS.
     = 0002      SELKEY   = 2
     = 0004      OPTKEY   = 4
     = 0007      ANYKEY   = STRTKY+SELKEY+OPTKEY

     = 0001      TXSL     = 1              ; SCREEN MODE = TEXT, SMALL LETTERS.
     = 0002      TXML     = 2              ; SCREEN MODE = TEXT, MEDIUM OR LARGE LETTERS.
     = 0004      GRSS     = 4              ; SCREEN MODE = GRAPHICS, SPLIT.
     = 0008      GRFS     = 8              ; SCREEN MODE = GRAPHICS, FULL.

                 ; 'NAME' TYPES.

     = 0080      ATRSTR   = $80            ; 'STRING' VARIABLE.
     = 0040      ATRNUM   = $40            ; 'NUMERIC' VARIABLE.
     = 0020      ATRIO    = $20            ; 'I/O' DEVICE.
     = 0000      ATRLIN   = 0              ; STATEMENT 'LINE'.
```

```
                    ; RESERVED COMMAND 'TOKENS' AND 'USRTAB' SIZE.

    = 00FE          TKNCNT  = $FE           ; COMMAND CONTINUATION.
    = 00FF          TKNNUL  = $FF           ; NULL COMMAND.

                    ; ROBOT TURTLE DRIVER COMMANDS.

    = 0000          RBOFF   = 0             ; 'ROBOT OFF'.
    = 0020          RBON    = $20           ; 'ROBOT ON'.
    = 0001          RBEYES  = 1             ; 'EYES'.
    = 0002          RBPEN   = 2             ; 'RPEN'.
    = 0003          RBHORN  = 3             ; 'HORN'.
    = 0080          RBFWD   = $80           ; 'GO +'.
    = 0081          RBBACK  = $81           ; 'GO -'.
    = 0040          RBLEFT  = $40           ; 'TURN +'.
    = 0041          RBRGHT  = $41           ; 'TURN -'.

                    ; LOAD TYPES.

    = 0001          KLOAD   = 1
    = 0002          KMERGE  = 2
    = 0003          KAPPND  = 3
```

```
                        ;
                        ; PILOT DATA BASE
                        ;
0000  = 0080                    ORG     $0080
      = 0080            DTAB    = *             ; BASE ADDRESS FOR DXXXI UTILITIES & OTHERS.

0080  = 0004            INLN    DS 4            ; INPUT LINE POINTER.
0084  = 0002            NXTLN   DS 2            ; NEXT LINE POINTER (RUN MODE).
0086  = 0001            ACOLR2  DS 1            ; AUTO-NUMBER COLOR REGISTER 2.
0087  = 0001            ACOLR1  DS 1            ; AUTO-NUMBER COLOR REGISTER 1.
0088  = 0004            ACLN    DS 4            ; ACCEPT LINE POINTER.
008C  = 0004            TELN    DS 4            ; TEXT EXPRESSION RESULT POINTER.

0090                    TABADR
0090  = 0002            TBLBAS  DS 2            ; COMMAND TABLE POINTER.

0092  = 0001            EXEC    DS 1            ; 0 = SYNTAX CHECK, ELSE EXECUTE (FOR X-ROUTINES).

0093  = 000E            EXPSTK  DS ESTKSZ       ; EXPRESSION STACK.
00A1  = 0006            TEMP    DS 6            ; TEMPORARY STORAGE FOR BOTTOM LEVEL ROUTINES.
00A7  = 0004            TEMP2   DS 4            ; MORE TEMPORARY STORAGE.
00AB  = 0003            XTEMP   DS 3            ; TEMPORARY STORAGE FOR X-ROUTINES.

00AE  = 0002            S1L     DS 2            ; DYNAMIC STORAGE AREA LIMITS.
00B0  = 0002            S1H     DS 2
00B2  = 0002            S2L     DS 2
00B4  = 0002            S2H     DS 2

00B6  = 0002            POINT   DS 2            ; 'ATOM' RETURN PARAMETER & 'PSF' WORK POINTER.
00B8  = 0002            NUMBER  DS 2            ; 'ATOM' RETURN PARAMETER & 'PSTOP' ERROR # SAVE.

00BA  = 0004            LP      DS 4            ; STRING PACKAGE LIST POINTER.
00BE  = 0004            NP      DS 4            ;                NAME POINTER.
00C2  = 0004            DP      DS 4            ;                DATA POINTER.
00C6  = 0004            MP      DS 4            ;                PATTERN MATCH POINTER.
00CA  = 0004            SP      DS 4            ;                SOURCE POINTER (BOTTOM LEVEL).
00CE  = 0004            PP      DS 4            ;                PATTERN POINTER (BOTTOM LEVEL).

00D2  = 0002            MEMA    DS 2            ; MEMORY MANAGEMENT ADDRESS PARAMETER.
00D4  = 0002            MEMB    DS 2            ;                   BYTE COUNT PARAMETER.
00D6  = 0002            MSP     DS 2            ;                   SOURCE POINTER.
00D8  = 0002            MDP     DS 2            ;                   DESTINATION POINTER.
00DA  = 0002            MBC     DS 2            ;                   WORKING BYTE COUNT.

00DC  = 0002            LINENO  DS 2            ; STATEMENT LINE # (MUST BE IN ZERO PAGE).
00DE  = 0002            LS      DS 2            ; 'XLIST' START LINE #, 'XGRAPH' ITERATION COUNT & 'SCNDEV'.
00E0  = 0002            LEND    DS 2            ; 'XLIST' END LINE #, 'GMOVE' REGISTER SAVE & 'SCNDEV'.
00E2                    MFDEL                   ; MATCH FIELD DELIMITER (',' OR  '!'.
00E2  = 0002            ACC     DS 2            ; WORKING NUMERIC ACCUMULATOR.
00E4  = 0002            IOSTAT  DS 2            ; COLLEEN I/O ERROR STATUS (WORD).

00E6  = 0003            GXNEW   DS 3            ; GRAPHICS NEXT POSITION (LSB,MSB,FRACTION).
00E9  = 0003            GYNEW   DS 3            ;
00EC  = 0003            GX      DS 3            ; GRAPHICS X POSITION (LSB,MSB,FRACTION).
00EF  = 0003            GY      DS 3            ; GRAPHICS Y POSITION (LSB,MSB,FRACTION).
00F2  = 0002            THETA   DS 2            ; POLAR ANGLE.
```

```
00F4  = 0002       FSTACK  DS 2            ; FLOOD STACK POINTER, & 'SETCLR' TEMP.
00F6  = 0002       ADRESS  DS 2            ; FLOOD SCREEN POINTER 'SSAVE', 'SLOAD' & 'NEWDRW' TEMP.
00F8  = 0002       TRADDR  DS 2            ; TURTLE REP. ADDRESS FOR VBLANK PROCESS.
00FA  = 0002       ALINE   DS 2            ; AUTO-INPUT & RENUMBER LINE NUMBER & TEMP.
00FC  = 0002       AINC    DS 2            ; AUTO-INPUT  & RENUMBER LINE INCREMENT.
00FE  = 0001       MATCHF  DS 1            ; MATCH RESULT (0 = FALSE, ELSE MATCH FIELD #).
00FF  = 0001       RUN     DS 1            ; 0 = IMMEDIATE MODE, ELSE RUN MODE.

                   ; REDEFINES OF VARIABLES FOR GRAPHICS USE

      = 00BE       GX1     = NP            ; END     X  [3 BYTES].
      = 00C1       GY1     = GX1+3         ; END     Y  [3 BYTES].
      = 00C4       GX2     = GY1+3         ; START   X  [3 BYTES].
      = 00C7       GY2     = GX2+3         ; START   Y  [3 BYTES].
      = 00CA       DELX    = GY2+3         ; DELTA   X  [2 BYTES].
      = 00CC       DELY    = DELX+2        ;         Y  [2 BYTES].
      = 00CE       GACC    = DELY+2        ; WORKING ACCUMULATOR [4 BYTES].
      = 00D2       GTEMP   = GACC+4        ;          TEMP  [4 BYTES].
      = 00D6       GTEMP2  = GTEMP+4       ;          TEMP  [4 BYTES].
      = 0093       DELTAR  = EXPSTK        ; DRAW DELTA Y[2].
      = 0095       DELTAC  = DELTAR+2      ; DRAW DELTA X[2].
      = 0097       ROWINC  = DELTAC+2      ; FILL Y INC. [1].
      = 0098       ROWAC   = ROWINC+1      ; DRAW Y ACC. [2].
      = 009A       COLAC   = ROWAC+2       ; DRAW X ACC. [2].
      = 009C       COUNTR  = COLAC+2       ; DRAW COUNTER [2].
      = 009E       ENDPT   = COUNTR+2      ; DRAW E [2].

                   ; REDEFINES OF 'EXPSTK' FOR EDIT COMMANDS.

      = 0093       BLOW    = EXPSTK        ; LOW BRACKET ADDRESS.
      = 0095       BHIGH   = EXPSTK+2      ; HIGH.
      = 0097       BNUM    = EXPSTK+4      ; # OF LINES IN RANGE.
      = 0099       RTMP    = EXPSTK+6      ; RENUMBER TEMP.
      = 009B       R2TMP   = EXPSTK+8      ; ".


0100  = 0500               ORG     $0500

0500  0000         USRTAB  DW 0            ; USER EXTENDABLE COMMAND TABLE.
                                           ; (MSBYTE = 0 IF UNUSED).
0502  0000         RBVECT  DW 0            ; ADDRESS OF ROBOT TURTLE DRIVER.
                                           ; (MSBYTE = 0 IF UNUSED).
0504  = 0002       IOEDIS  DS 2            ; I/O ERROR STOP DISABLE.
                                           ; EXTRA BYTE TO PROTECT AGAINST WORD POKE.
0506  = 0001       EXECF   DS 1           ; CONDITION RESULT (0 = NO EXECUTE, ELSE EXECUTE).
0507  = 0003       XJUMP   DS 3            ; FIRST BYTE = JMP COMMAND (X-ROUTINES).
050A  = 0003       GJUMP   DS 3            ; FIRST BYTE = JMP COMMAND (G-ROUTINES).
050D  = 0003       SJUMP   DS 3            ; FIRST BYTE = JMP COMMAND ('SOP').
0510  = 0001       CTABAT  DS 1           ; 'ATTMBYTE' BYTE FROM 'CMATCH'.

0511  = 0001       DIGIT   DS 1
0512  = 0001       SAVYR   DS 1           ; 'MLOOP' SAVE Y REGISTER.
0513  = 0001       PEN     DS 1           ; GRAPHICS PEN SELECT.
0514  = 0001       GRFLAG  DS 1           ; GRAPHICS MODE FLAG (0=NOT GRAPHICS, ELSE GRAPHICS).
0515  = 0008       AUDIOR  DS AUREGS+AUREGS ; AUDIO VARIABLE POINTERS.
051D  = 0001       AUX1    DS 1           ; I/O AUX1 OVERRIDE BYTE.
```

```
                    ATARI CAMAC Assembler Ver  1.0A  Page   9
PILOT -- R.B. STEWART                                   D1:PILOT.

0051E  = 0001       AUX2    DS 1            ; I/O AUX2 OVERRIDE BYTE.
051F  = 0001                DS 1            ; 'OPNBUF'-1 USED BY 'SCNDEV'.
0520  = 0010       OPNBUF  DS DNSIZE+1     ; DEVICE NAME BUFFER FOR OPEN.
0530  = 0002       CDEST   DS 2            ; 'CHOT' DESTINATION IDENTIFIER & SAVE BYTE.
0532  = 0001       LOADFG  DS 1            ; 0 IF NOT LOADING, ELSE LOADING.
0533  = 0002       MATCHX  DS 2            ; 'XMATCH' FIELD INDEX VALUES.
0535  = 0001       TRACE   DS 1            ; RUN-TIME TRACE FLAG (TRACE IF <> 0).
0536  = 0001       AUTOIN  DS 1            ; AUTO-INPUT FLAG (ACTIVE IF <> 0).

0537  = 0001       GSMODE  DS 1            ; GRAPHICS SCREEN MODE.
0538  = 000A       INLNBF  DS INBFSZ       ; TEMP STORAGE FOR SOURCE TO MATCH.
0542  = 0001       NAMLNG  DS 1            ; 'SAVIT' & 'RESIT'.
0543  = 0001       NOCONT  DS 1                        ; 0 IF CONTINUE O.K.
0544  = 0001       CONKEY  DS 1                        ; 1=START, 2=SELECT, 4=OPTION.
0545  = 0001       SGLSTP  DS 1                        ; SINGLE STEP IF .NE. 0.
0546  = 0001       AXFLAG  DS 1                        ; 1 IF ACCEPT LITERAL.
0547  = 0001       AKFLAG  DS 1                        ; 1 IF ACCEPT KEY.
0548  = 0001       XXXX    DS 1                        ; 'SCNDEV' & 'PSTOP' USE.
0549  = 0004       GNUMB   DS 4                        ; GRAPHICS WORKING STORAGE & "XACCPT' TEMPORARY.
054D  = 0001       USTKP   DS 1                        ; USE STACK POINTER (0 - N*2).
054E  = 0001       ESTKP   DS 1                        ; EXPRESSION STACK POINTER.
054F  = 0001       TRTLON  DS 1                        ; 0=VISIBLE TURTLE OFF, ELSE ON.
0550  = 0001       TRTSNS  DS 1                        ; VISIBLE TURTLE SENSOR STATE.
0551  = 0001       LETTRSZ DS 1                        ; TEXT LETTER SIZE: 0,1 OR 2.
0552  = 0001       SPLTSC  DS 1                        ; 0=FULL GRAPHICS, $10 = SPLIT SCREEN.
      = 000A       NMBFSZ  =  5*2                      ; 'MNYNMS' BUFFER SIZE.
0553  = 000A       NMSBF   DS NMBFSZ
055D  = 0001       SPEED   DS 1                        ; SPEED CONTROL.
055E  = 0001       EDGRUL  DS 1                        ; TURTLE EDGE RULE.
055F  = 0001       TRYPOS  DS 1                        ; VISIBLE TURTLE Y POSITION.
0560  = 0001       ORIENT  DS 1                        ; VISIBLE TURTLE ORIENTATION.
0561  = 0002       XC      DS 2                        ; SCREEN CENTER X.
0563  = 0002       YC      DS 2                        ; SCREEN CENTER Y.
0565  = 0001       CSTATE  DS 1                        ; CONSOLE KEY READ STATE.
0566  = 0001       ATRTYP  DS 1                        ; 'NAME' ATTRIBUTE FOR 'IFIND'.
0567  = 0001       DMPTYP  DS 1                        ; ATTRIBUTE FOR DUMP CODE.
0568  = 0001       TKNTYP  DS 1                        ; TOKENIZED COMMAND
0569  = 0001       LSTKN   DS 1                        ; TOKEN FROM PREVIOUS STATEMENT FOR ': CONTINUATION'.
056A  = 0001       TKNOFF  DS 1                        ; OFFSET PAST COMMAND.
056B  = 0030       USESTK  DS USTKSZ                   ; USE STACK.
059B  = 0001       FCOLOR  DS 1                        ; 'FLOOD' COLOR
059C  = 0001       FLDCLR  DS 1                        ; FIELD COLOR TO BE FLOODED.
059D  = 0001       MSKTMP  DS 1                        ; TEMP MASKED DATA.
059E  = 0001       ROWFLG  DS 1                        ; ROW FLAG.
059F  = 0001       COLFLG  DS 1                        ; COLUMN FLAG.
05A0  = 0001       SAVROW  DS 1                        ; SAVED STARTING ROW.
05A1  = 0002       SAVCOL  DS 2                        ; SAVED STARTING COLUMN.
05A3  = 0002       LFTCOL  DS 2                        ; LEFT COLUMN VALUE.
05A5  = 0002       NEWLC   DS 2                        ; NEW LEFT COLUMN.
05A7  = 0002       RGTCOL  DS 2                        ; RIGHT COLUMN VALUE.
05A9  = 0002       NEWRC   DS 2                        ; NEW RIGHT COLUMN.
05AB  = 0001       MAXROW  DS 1                        ; MAXIMUM ROW VALUE.
05AC  = 0002       MAXCOL  DS 2                        ; MAXIMUM COLUMN VALUE.
05AE  = 0002       MLTTMP  DS 2                        ; MULTIPLY TEMP.
05B0  = 0001       SHFAMT  DS 1                        ; SHIFT AMOUNT.
05B1  = 0001       DMASK   DS 1
05B2  = 0001       FINEFG  DS 1                        ; 0 = COARSE SCROLL, -1 = FINE.
```

```
                ATARI CAMAC Assembler Ver  1.0A  Page  10
PILOT -- H.R. STEWART                                       D1:PILOT.


05B3  = 0002       CETEMP  DS 2                 ; 'COMPRS'/'EXPAND' TEMPORARY.
05B5  = 0001       LFCOL   DS 1                 ; T: LEFT MOST COLUMN.
05B6  = 0001       RGCOL   DS 1                 ; T: RIGHT MOST COLUMN.
05B7  = 0001       PENNUM  DS 1                 ; PEN NUMBER.
05B8  = 0001       PENCOL  DS 1                 ; PEN COLOR.
05B9  = 0001       NCOLRS  DS 1                 ; NUMBER OF COLORS ALLOWED.
05BA  = 0001       NXTCLR  DS 1                 ; NEXT AVAILABLE COLOR SLOT.
05BB               PNCLRS                       ; PEN COLORS.
05BB  = 0001       BAKCLR  DS 1                 ; BACKGROUND COLOR.
05BC  = 0008               DS 8                 ; FOREGROUND COLORS.
05C4  = 0001       TRTCOL  DS 1                 ; TURTLE COLOR.
05C5  = 0001       RBTON   DS 1                 ; 0=ROBOT TURTLE OFF, ELSE ON.
05C6  = 0001       RBTSNS  DS 1                 ; ROBOT SENSOR STATE.
05C7  = 0001       RBTCMD  DS 1                 ; INTERNAL ROBOT COMMAND.
05C8  = 0002       RBTPRM  DS 2                 ; INTERNAL ROBOT PARAMETER.
05CA  = 0001       INDENT  DS 1                 ; AUTO INDENT.
05CB  = 0001       SCTEMP  DS 1                 ; 'SETCLR' TEMP.
05CC  = 0001       PRTEMP  DS 1                 ; 'PRCLNM' TEMP.
05CD  = 0002       WALLS   DS 2                 ; WALL SELECTION DATA.
05CF  = 0002       GCOL    DS 2                 ; TURTLE COLUMN POSITION.
05D1  = 0001       GROW    DS 1                 ; TURTLE ROW POSITION.
05D2  = 0002       GANGLE  DS 2                 ; TURTLE THETA.
05D4  = 0001       GROPR   DS 1                 ; GRAPHICS OPERATION TYPE.
05D5  = 0001       HITWLL  DS 1                 ; 0 = NO WALL HIT, ELSE WALL HIT.
05D6  = 0001       HITEDG  DS 1                 ; 0 = NO EDGE HIT, ELSE EDGE HIT.
05D7  = 0001       HALTFG  DS 1                 ; NON-ZERO = HALT AT EDGE.
05D8  = 0001       GWTEMP  DS 1                 ; 'GREAD' TEMP.
05D9  = 0001       TUFLAG  DS 1                 ; TURTLE PARAMETER UPDATE INTERLOCK.
05DA  = 0001       LITMAT  DS 1                 ; NON-ZERO = LITERAL MATCH.
05DB  = 0001       NOPLOT  DS 1                 ; NON-ZERO = DON'T PUT POINT.
05DC  = 0002       DSISAV  DS 2                 ; VALUE OF ORIGINAL 'DOSINI'.
05DE  = 0002       DSVSAV  DS 2                 ; VALUE OF ORIGINAL 'DOSVEC'.
05E0  = 0001       DMASAV  DS 1                 ; 'DMACT' SAVE VALUE FOR TV ON/OFF.
      = 011F       SPARES  = $700-*             ; *** THIS HAD BETTER BE POSITIVE ***


05E1  = BC00               ORG $BC00
BC00  = 00FF       TEXBUF  DS TEXLNG+1   ; TEXT EXPRESSION BUFFER.
BCFF  = 0001               DS 1          ; ONE EXTRA LEADING BLANK FOR AUTO-IN.
BD00  = 0078       COMBUF  DS LINLNG+1   ; COMMAND INPUT BUFFER.
BD78  = 0100       ACCBUF  DS 256        ; ACCEPT BUFFER.
BE78  = 0101       NAMBUF  DS 257        ; STRING NAME BUFFER.
```

```
                        ;
                        ; NOTE: THE USE OF THE TERM '(BRA)' IN A COMMENT INDICATES THAT THE
                        ; PARTICULAR BRANCH INSTRUCTIONS USED WILL ALWAYS BRANCH IN THE
                        ; PARTICULAR CIRCUMSTANCES.  THE BRANCH IS SUPPOSED TO BE A TWO BYTE
                        ; JUMP.
                        ;

                        ; POWER-UP ROUTINE AND INITIALIZATION

BF7C  = 7700                    ORG $7700

7700                    PILLOW                          ; PILOT LOW ADDRESS
7700                    TPBUFF                          ;
      = 770C            TVBUFF  = TPBUFF+12             ; VISIBLE REGION.
      = 7705            TRBUFF  = TVBUFF-7              ; INCLUDES UNDERFLOW.

7700  = 7800                    ORG TPBUFF+256          ; TURTLE REP. BUFFER.
```

```
       = 0000               IF      DOS
        -                   LDA     #0              ; CLEAR COLDSTART FLAG (SEE 'XDOS').
        -                   STA     COLDST
        -                   JMP     MLE             ; RETURN FROM DOS.
                            ENDIF

 7800                       PROC
 7800  A508         PILINI  LDA     WARMST          ; WARM START?
 7802  D015 ^7819           BNE     :PI020          ; YES.

 7804  A50C                 LDA     DOSINI          ; SAVE ORIGINAL 'DOSINI'.
 7806  8DDC05               STA     DSISAV
 7809  A50D                 LDA     DOSINI+1
 780B  8DDD05               STA     DSISAV+1

 780E  A900                 LDA     # LOW PILINI    ; PLUG IN NEW 'DOSINI'.
 7810  850C                 STA     DOSINI
 7812  A978                 LDA     # HIGH PILINI
 7814  850D                 STA     DOSINI+1

       = 0000               IF      DOS
        -                   LDA     DOSVEC          ; SAVE ORIGINAL 'DOSVEC'.
        -                   STA     DSVSAV
        -                   LDA     DOSVEC+1
        -                   STA     DSVSAV+1
                            ENDIF

 7816  4C1C78               JMP     :PI030

 7819  202578       :PI020  JSR     GODOS           ; PERFORM DOS INIT.

 781C  A9C3         :PI030  LDA     # LOW MLE       ; CHANGE 'DOSVEC' FOR PILOT ENTRY.
 781E  850A                 STA     DOSVEC
 7820  A976                 LDA     # HIGH MLE
 7822  850B                 STA     DOSVEC+1
 7824  60                   RTS

 7825  6CDC05       GODOS   JMP     (DSISAV)        ; 2ND HALF OF JSR (DSISAV).
```

```
7828                        PROC
7828  A906          INIT    LDA     #EPUTC-IOVBAS    ; ESTABLISH 'CHOT' DESTINATION AS 'E:'.
782A  8D3C05                STA     CDEST

782D  A508                  LDA     WARMST           ; WARM START?
782F  D037 ^7868            BNE     :INI15           ; YES.

7831  A280                  LDX     #$80             ; CLEAR UPPER HALF OF PAGE ZERO.
7833  A900                  LDA     #0

7835  9500          :INI10  STA     0,X
7837  E8                    INX
7838  D0FB ^7835            BNE     :INI10           ; CONTINUE TILL PAGE WRAP POINT.

783A  A94C                  LDA     #$4C             ; PUT JMP OP-CODE IN JUMP VECTORS.
783C  8D0705                STA     XJUMP+0
783F  8D0A05                STA     GJUMP+0
7842  8D0D05                STA     SJUMP+0

7845  A916                  LDA     #$16             ; AUTO-NUMBER SCREEN = DARK YELLOW.
7847  8586                  STA     ACOLR2
7849  A900                  LDA     #$00             ; AUTO-NUMBER LETTERS = BLACK.
784B  8587                  STA     ACOLR1

        = 0000              IF      FALSE
          -                 LDA     # LOW PRGEND     ; SET 'MEMLO' AFTER END OF PROGRAM.
          -                 STA     MEMLO
          -                 LDA     # HIGH PRGEND
          -                 STA     MEMLO+1
                            ENDIF

784D  ADE702                LDA     MEMLO            ; ESTABLISH MEMORY LIMITS FOR ALLOCATION.
7850  85AE                  STA     S1L              ; ... & PROGRAM STORAGE AREA.
7852  ADE802                LDA     MEMLO+1
7855  85AF                  STA     S1L+1
7857  20C087                JSR     CLRPRG

785A  ADE502                LDA     MEMHI            ; ALSO FOR STRING STORAGE AREA.
785D  85B4                  STA     S2H
785F  85B2                  STA     S2L
7861  ADE602                LDA     MEMHI+1
7864  85B5                  STA     S2H+1
7866  85B3                  STA     S2L+1

7868  A284          :INI15  LDX     # HIGH PILVBL    ; INTERCEPT VBLANKS.
786A  A091                  LDY     # LOW PILVBL
786C  A907                  LDA     #7               ; VVBLKD.
786E  205CE4                JSR     SETVBV

7871  A900                  LDA     #0               ; ZERO ...
7873  8D3205                STA     LOADFG           ; ... LOAD FLAG ...
7876  8D3505                STA     TRACE            ; ... TRACE FLAG ...
7879  8D3605                STA     AUTOIN           ; ... AUTO-INPUT FLAG ...
787C  8D1D05                STA     AUX1             ; ... I/O AUX1 ...
787F  8D1E05                STA     AUX2             ; ... I/O AUX2 ...
7882  85FE                  STA     MATCHF           ; ... & MATCH RESULT.
7884  8D1D05                STA     CTABAT           ; ... LAST COMMAND ATTRIBUTES.
```

```
7887  8DC505               STA    RBTON           ; ... ROBOT TURTLE.
788A  8D5005               STA    SPEED           ; FULL SPEED.
788D  8DE205               STA    FINEFG          ; COARSE SCROLLING.
7890  8DF005               STA    DMASAV

7893  A976                 LDA    # LOW ACCBUF    ; SET ACCEPT BUFFER POINTER.
7895  8588                 STA    ACLN
7897  A98D                 LDA    # HIGH ACCBUF
7899  8589                 STA    ACLN+1
789B  20729F               JSR    NULACC          ; SET ACCEPT BUFFER TO NULL.

789E  A900                 LDA    # LOW TEXBUF    ; SETUP TEXT EXPRESSION BUFFER POINTER.
78A0  858C                 STA    TELN
78A2  A98C                 LDA    # HIGH TEXBUF
78A4  858D                 STA    TELN+1
78A6  A920                 LDA    #' '            ; LEADING BLANK FOR AUTO-IN.
78A8  8DFFBC               STA    COMBUF-1

78AB  20E1A5               JSR    TRTINI          ; INITIALIZE VISIBLE TURTLE STUFF.
78AE  2050B3               JSR    RBINIT          ; ... ROBOT TURTLE ('OFF' IN 'MLE').
78B1  209E98               JSR    REMDEV          ; REMOVE DEVICE ASSIGNMENTS FROM STRING LIST.
78B4  20F494               JSR    TXOPEN          ; OPEN E: & RECAPTURE GRAPHICS REGION IF NECESSARY.

78B7  A508                 LDA    WARMST          ; WARMSTART?
78B9  D005 ^78C0           BNE    :INI30          ; YES.

78BB  A917                 LDA    #SIGNON         ; GENERATE SIGN-ON MESSAGE.
78BD  20FFB4               JSR    MESSOT

78C0  4C2CB5        :INI30 JMP    RDYMES          ; GENERATE "READY" MESSAGE & RETURN.
```

```
 78C3                     PROC
                    ;
                    ; MAIN LOOP FOR PILOT INTERPRETER.
                    ;
                    ;
                    ; POWER-UP AND RESET ENTRY.
                    ;
 78C3  A2FF         MLE      LDX      #$FF             ; INITIALIZE STACK POINTER.
 78C5  9A                    TXS

 78C6  A970                  LDA      # HIGH $7000     ; NEW TOP OF MEMORY.
 78C8  856A                  STA      RAMTOP
 78CA  A900                  LDA      #0               ; CLEAR ESSENTIALS FOR EOPEN CALL.
 78CC  8D1D05                STA      AUX1
 78CF  8D1E05                STA      AUX2
 78D2  8D1405                STA      GRFLAG
 78D5  8D5105                STA      LETTRSZ
 78D8  8DB205                STA      FINEFC
 78DB  20FE96                JSR      EOPEN            ; MOVE SCREEN DOWN.

 78DE  8E4305                STX      NOCCNT           ; NO CONTINUATION.

 78E1  202878                JSR      INIT             ; INITIALIZE REST OF ENVIRONMENT.

                    ; *** EXTERNAL ENTRY POINT ***

 78E4  A93C         MLRES    LDA      #CASSOF          ; CASSETTE MOTOR OFF
 78E6  8D02D3                STA      PACTL
 78E9  20849F                JSR      AUDCLR           ; CLEAR AUDIO REGISTERS

 78EC  A900         MLRES2   LDA      #0               ; RESET ...
 78EE  85FF                  STA      RUN              ; ... RUN FLAG ...
 78F0  8D4505                STA      SGLSTP           ; ... SINGLE STEP ...
 78F3  8DFE02                STA      DSPFLG           ; ... DISPLAY FLAG ...
 78F6  8DB602                STA      INVFLG           ; ... INVERT VIDEO FLAG ...
 78F9  8D0405                STA      IOEDIS           ; ... & ERROR STOP DISABLE FLAG.

 78FC  A9E1                  LDA      # LOW XTYPE      ; MAKE ':' COMMAND = 'T:'.
 78FE  8D0805                STA      XJUMP+1
 7901  A9F3                  LDA      # HIGH XTYPE
 7903  8D0905                STA      XJUMP+2
 7906  8D0605                STA      EXECF            ; CONDITION FLAG = TRUE.

                    ; *** EXTERNAL ENTRY POINT ***

 7909  A900         MLLOAD   LDA      # LOW COMBUF     ; RE-ESTABLISH CONSOLE BUFFER INPUT.
 790B  8580                  STA      INLN
 790D  A98D                  LDA      # HIGH COMBUF
 790F  8581                  STA      INLN+1

 7911  20007B       MLOOP    JSR      GETCOM           ; GET A COMMAND INPUT.
 7914  D07A ^7990            BNE      :MLO90           ; ERROR (SKIP BRANCH).

                    ; NOTE: THE Y REGISTER IS ASSUMED TO CONTAIN THE INDEX TO 'INLN'
                    ;       THROUGHOUT THIS ROUTINE.  ALL CALLED ROUTINES WILL BE
                    ;       RESPONSIBLE FOR MAINTAINING ITS INTEGRITY.
```

```
7916  AD3205            LDA    LOADFG          ; LOADING?
7919  D00C ^7927        BNE    :ML020          ; YES.

791B  A5FF              LDA    RUN             ; RUN MODE?
791D  D03B ^795A        BNE    :ML070          ; YES.

791F  AD3605            LDA    AUTOIN          ; AUTO-INPUT MODE?
7922  F003 ^7927        BEQ    :ML020          ; NO.

7924  4C9D79            JMP    :ML100          ; YES.

7927  AD4405    :ML020  LDA    CONKEY          ; CONSOLE KEY PRESSED?
792A  2901              AND    #STRTKY         ; START KEY?
792C  F017 ^7945        BEQ    :ML030          ; NO.

792E  AD4405            LDA    CONKEY          ; RESET THE KEY FLAG.
7931  29FE              AND    #$FF-STRTKY
7933  8D4405            STA    CONKEY

7936  AD4305            LDA    NOCONT          ; CAN WE CONTINUE?
7939  F003 ^793E        BEQ    :ML025          ; YES.

793B  201185            JSR    XPN010          ; NO -- START AT BEGINNING.

793E  CE4505    :ML025  DEC    SGLSTP          ; YES -- SET FLAG.
7941  C6FF              DEC    RUN             ; SET TO RUN MODE.
7943  D0CC ^7911        BNE    MLOOP           ; (BRA).

7945  20D39E    :ML030  JSR    SCNLBL          ; SCAN OVER LABEL IF PRESENT.
7948  F007 ^7951        BEQ    :ML040          ; YES -- SAW A VALID LABEL.

794A  B180              LDA    (INLN),Y        ; CHECK FOR LINE NUMBER.

794C  20B39E            JSR    CNUMBR          ; NUMBERED LINE?
794F  905E ^79AF        BCC    :ML110          ; YES -- EDIT MODE.

                ; UN-NUMBERED LINE -- IMMEDIATE EXECUTION

7951  A240      :ML040  LDX    #CTIMM          ; SETUP FOR IMMEDIATE MODE COMMANDS.
7953  20AB7B            JSR    SYCMND          ; IMMEDIATE MODE -- SYNTAX CHECK CODE.
7956  D038 ^7990        BNE    :ML090          ; ERROR -- DON'T EXECUTE THE COMMAND.

7958  F033 ^798D        BEQ    :ML085          ; (BRA).

                ; LINE FROM STORAGE -- 'RUN' MODE

795A  AD4405    :ML070  LDA    CONKEY          ; CONSOLE KEY PRESSED?
795D  2904              AND    #OPTKEY         ; OPTION KEY?
795F  F010 ^7971        BEQ    :ML080          ; NO.

7961  AD3505            LDA    TRACE           ; YES -- TOGGLE THE TRACE.
7964  4901              EOR    #KON
7966  8D3505            STA    TRACE

7969  AD4405            LDA    CONKEY          ; RESET THE KEY FLAG.
796C  29FB              AND    #$FF-OPTKEY
796E  8D4405            STA    CONKEY
```

```
7971  AD3505      :ML080  LDA     TRACE            ; TRACE EXECUTION?
7974  0D4505              ORA     SGLSTP
7977  F014 ^798D          BEQ     :ML085           ; NO.

7979  20B896              JSR     TSTMOD           ; CHECK SCREEN MODE.
797C  2905                AND     #TXSL+GRSS       ; TEXT OUPUT O.K.?
797E  D003 ^7983          BNE     :ML082           ; YES.

7980  20F494              JSR     TXOPEN           ; NO -- OPEN TEXT SCREEN.

7983  A918        :ML082  LDA     #TRCMES          ; PRINT TRACE LINE HEADER.
7985  20FFB4              JSR     MESSOT
7988  A000                LDY     #INLN-DTAB       ; PRINT SOURCE STATEMENT.
798A  20229F              JSR     PSF

                  ; COMMON CODE 'IMMEDIATE' AND 'RUN'

798D  20E27B      :ML085  JSR     EXCMND           ; EXECUTE THE COMMAND.
7990  D06E ^7A00  :ML090  BNE     :ML155           ; RUN-TIME ERROR (SKIP BRANCH POINT).

7992  AD4505              LDA     SGLSTP           ; SINGLE STEP?
7995  F003 ^799A          BEQ     :ML095           ; NO.

7997  4CEC78              JMP     MLRES2           ; YES -- RETURN TO IMMEDIATE MODE.

799A  4C1179      :ML095  JMP     MLOCP            ; GET NEXT COMMAND.

                  ; AUTO-INPUT MODE -- SUPPLY THE LINE NUMBER AND ONE EXTRA LEADING BLANK.

799D  A5FA        :ML100  LDA     ALINE            ; SUPPLY THE LINE NUMBER.
799F  85B8                STA     NUMBER
79A1  A5FB                LDA     ALINE+1
79A3  85B9                STA     NUMBER+1

79A5  A200                LDX     #INLN-DTAB
79A7  20129D              JSR     DCCRI            ; ONE EXTRA LEADING BLANK.
79AA  E683                INC     INLN+3

79AC  4C8279              JMP     :ML112

                  ; NUMBERED LINE INPUT -- EDIT MODE.

79AF  20BE81      :ML110  JSR     ATON             ; CONVERT LINE NUMBER TO BINARY IN 'NUMBER'.

79B2  8482        :ML112  STY     INLN+2           ; SAVE INPUT LINE POINTER.

79B4  A238                LDX     #NUMBER-DTAB
79B6  AD3205              LDA     LOADFG           ; SUPPLY LINE NUMBER IF 'APPEND'.
79B9  C903                CMP     #KAPPND
79BB  D005 ^79C2          BNE     :ML120

79BD  A074                LDY     #ALINE-DTAB
79BF  20459A              JSR     [illegible]

79C2  20D59A      :ML120  JSR     CHKLN            ; CHECK LINE # FOR RANGE.
79C5  B05F ^7A26          BCS     :ML200           ; OUT OF RANGE.
```

```
79C7  A900                  LDA    #0              ; CLEAR USE STACK ON INSERT/DELETE.
79C9  8D4D05                STA    USTKP

79CC  A9FF                  LDA    #$FF            ; ALTER PROGRAM -- NO CONTINUATION.
79CE  8D4305                STA    NOCONT

79D1  A589                  LDA    NUMBER+1        ; SAVE LINE NUMBER ...
79D3  85DC                  STA    LINENO          ; ... IN INVERTED FORM (STRING NAME).
79D5  A588                  LDA    NUMBER
79D7  85DD                  STA    LINENO+1

79D9  A482                  LDY    INLN+2          ; RESTORE INPUT LINE INDEX.
79DB  20D39E                JSR    SCNLBL          ; SKIP OVER LABEL IF PRESENT.
79DE  F019 ^79F9            BEQ    :ML150          ; LABEL FOUND.

79E0  B180                  LDA    (INLN),Y        ; CHECK FOR NULL STATEMENT.
79E2  C99B                  CMP    #EOL
79E4  D013 ^79F9            BNE    :ML150          ; NON-NULL -- STATEMENT IS TO BE ENTERED.

79E6  AD3605                LDA    AUTOIN          ; AUTO-INPUT MODE?
79E9  F008 ^79F3            BEQ    :ML140          ; NO.

79EB  20167A                JSR    LVAUTO          ; LEAVE AUTO-INPUT MODE.

79EE  A981                  LDA    #AUTOXT         ; GENERATE MESSAGE AS WE LEAVE.
79F0  4CC57A                JMP    :ML985

79F3  20E77A       :ML140   JSR    LDELFT          ; YES -- DELETE NUMBERED LINE.
79F6  4C1179       :ML145   JMP    MLOOP

79F9  A482         :ML150   LDY    INLN+2          ; RESTORE INPUT LINE POINTER.

79FB  A220                  LDX    #CTRUN          ; SETUP FOR RUN MODE COMMANDS.
79FD  20AB7B                JSR    SYCMND          ; SYNTAX CHECK THE STATEMENT.
7A00  D038 ^7A3A   :ML155   BNE    :ML900          ; SYNTAX ERROR (SKIP BRANCH POINT).

7A02  20CE7A                JSR    LINSRT          ; INSERT THE NEW LINE (COMMAND 'TOKENIZED').
7A05  D01F ^7A26            BNE    :ML200          ; NO ROOM FOR NEW LINE.

7A07  A27A                  LDX    #ALINE-DTAB     ; INCREMENT AUTO-INPUT LINE #.
7A09  A07C                  LDY    #AINC-DTAB      ; (EVEN IF NOT IN AUTO-INPUT MODE).
7A0B  20329C                JSR    DADDI

7A0E  AD3605                LDA    AUTOIN          ; AUTO-INPUT MODE?
7A11  F0E3 ^79F6            BEQ    :ML145          ; NO -- GET NEXT COMMAND.
7A13  4C0979                JMP    MLLOAD          ; YES -- ADJUST 'INLN' FOR 'LEADING BLANK'.

7A16  A200         LVAUTO   LDX    #0              ; RESET AUTO-INPUT MODE.
7A18  8E3605                STX    AUTOIN

7A1B  A2B4                  LDX    #CBLUE          ; RESTORE NORMAL SCREEN COLOR.
7A1D  8EC602                STX    COLOR0+2
7A20  A21A                  LDX    #CYELLO
7A22  8EC502                STX    COLOR0+1
7A25  60                    RTS
```

```
                        ; NO ROOM FOR LINE OR LINE # OUT OF RANGE.

7A26  48               :ML200   PHA                        ; SAVE ERROR CODE.
7A27  AD3205                    LDA      LOADFG            ; LOAD IN PROGRESS?
7A2A  F00A ^7A36                BEQ      :ML210            ; NO.

7A2C  A900                      LDA      #0                ; ABORT LOAD.
7A2E  8D3205                    STA      LOADFG
7A31  A230                      LDX      #IOCB3            ; CLOSE FILE.
7A33  203F97                    JSR      DCLOSE

7A36  68               :ML210   PLA                        ; RESTORE ERROR CODE.

7A37  20167A                    JSR      LVAUTO            ; LEAVE AUTO-INPUT MODE & FALL INTO 'PSTOP'.
```

```
                   ; SYNTAX/RUN-TIME ERROR PROCESSOR

                   ; *** EXTERNAL ENTRY POINT ***

                   ;         A = ERROR CODE.
                   ;         Y = INDEX TO ERROR IN STATEMENT.

7A3A               :ML900
7A3A  A2FF         PSTOP   LDX     #$FF             ; RE-INIT STACK POINTER.
7A3C  9A                   TXS
7A3D  8EFC02               STX     DSPFLG           ; SET DISPLAY FLAG.

7A40  8C1205               STY     SAVYR            ; SAVE INDEX TO ERROR.
7A43  85B8                 STA     NUMBER           ; SAVE ERROR NUMBER.

7A45  20B896               JSR     TSTMOD           ; CHECK SCREEN MODE.
7A48  2905                 AND     #TXSL+GRSS       ; TEXT OUTPUT O.K.?
7A4A  D003 ^7A4F           BNE     :ML920           ; YES.

7A4C  20F494               JSR     TXOPEN           ; NO -- OPEN TEXT SCREEN.

7A4F  A906         :ML920  LDA     #EPUTC-IOVBAS    ; RE-ESTABLISH 'E:' AS 'CHOT' OUTPUT.
7A51  8D3005               STA     CDEST
7A54  20989F               JSR     NEWLIN

7A57  A5FF                 LDA     RUN              ; IF IMMEDIATE ...
7A59  0583                 ORA     INLN+3           ; ... & EMPTY INPUT LINE ...
7A5B  F06E ^7ACB           BEQ     :ML990           ; ... THEN IGNORE ERROR (BREAK).

7A5D  A5B8                 LDA     NUMBER
7A5F  C981                 CMP     #EOPERR          ; SEE IF ERROR IS END OF PROGRAM.
7A61  F062 ^7AC5           BEQ     :ML985           ; YES -- NO STATEMENT TO PRINT.

7A63  A8                   TAY                      ; (SET CC).
7A64  3012 ^7A78           BMI     :ML947           ; YES -- NO HIGHLIGHTED CHARACTER.

7A66  AC1205               LDY     SAVYR            ; HIGHLIGHT THE ERROR CHARACTER.
7A69  B180                 LDA     (INLN),Y
7A6B  8D4805               STA     XXXX             ; SAVE FOR LATER RESTORATION.
7A6E  C99B                 CMP     #EOL
7A70  D002 ^7A74           BNE     :ML945

7A72  A920                 LDA     #' '              ; REPLACE EOL WITH BLANK.

7A74  4980         :ML945  EOR     #$80             ; INVERT COLOR.
7A76  9180                 STA     (INLN),Y

7A78  A5FF         :ML947  LDA     RUN              ; SEE IF RUN OR IMMEDIATE MODE.
7A7A  F008 ^7A84           BEQ     :ML950           ; IMMEDIATE.

7A7C  A000                 LDY     #INLN-CTAB
7A7E  20229F               JSR     PSF              ; RUN -- PRINT STORAGE FORMAT.
7A81  4C8074               JMP     :ML960

7A84               :ML950
7A84  A900                 LDA     #0               ; *** OR DON'T USE 'INLN'+2 AS TEMP STORE ***
7A86  8582                 STA     INLN+2
```

```
7A88  A200                  LDX    #INLN-DTAB      ; IMMEDIATE -- PRINT INPUT LINE.
7A8A  209797                JSR    PRTSTG

7A8D  A588          :ML960  LDA    NUMBER          ; WAS THERE A HIGHLIGHTED CHARACTER?
7A8F  300F ^7AA0            BMI    :ML963          ; NO.

7A91  AC1205                LDY    SAVYR           ; RESTORE ORIGINAL CHARACTER.
7A94  AD4805                LDA    XXXX
7A97  9180                  STA    (INLN),Y
7A99  C99B                  CMP    #EOL            ; WAS IT THE EOL?
7A9B  D003 ^7AA0            BNE    :ML963          ; NO.

7A9D  208294                JSR    CHOT            ; YES -- DO IT NOW.

7AA0  A908          :ML963  LDA    #ASTMES         ; PREFIX MESSAGE WITH '***'.
7AA2  20FFB4                JSR    MESSOT

7AA5  A588                  LDA    NUMBER
7AA7  C986                  CMP    #IOERR          ; I/O ERROR?
7AA9  D00A ^7AB5            BNE    :ML981          ; NO.

7AAB  A4E4                  LDY    IOSTAT          ; YES -- BREAK?
7AAD  C080                  CPY    #128
7AAF  D004 ^7AB5            BNE    :ML981          ; NO.

7AB1  A987                  LDA    #ABTERR         ; YES -- CHANGE ERROR CODE.
7AB3  8588                  STA    NUMBER

7AB5  20FFB4        :ML981  JSR    MESSOT          ; GENERATE ERROR MESSAGE.
7AB8  A588                  LDA    NUMBER
7ABA  C986                  CMP    #IOERR          ; I/O ERROR?
7ABC  D005 ^7AC3            BNE    :ML982          ; NO.

7ABE  A264                  LDX    #IOSTAT-DTAB    ; YES -- PRINT ERROR STATUS.
7AC0  20149E                JSR    DECASC

7AC3  A908          :ML982  LDA    #ASTMES         ; APPEND '***' TO END OF MESSAGE.

                    ; *** EXTERNAL ENTRY POINT FROM 'MLOOP' ***

7AC5  20FFB4        :ML985  JSR    MESSOT
7AC8  20989F                JSR    NEWLIN

7ACB  4CE478        :ML990  JMP    MLRES           ; GET NEXT COMMAND.
```

```
                        ; LINE INSERT AND DELETE ROUTINES


7ACE                            PROC
                        ;
                        ; LINSRT -- INSERT NUMBERED LINE TO STATEMENT LIST
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       'LINENO' = LINE # (BINARY)
                        ;       'INLN' POINTS TO STATEMENT TO INSERT
                        ;       'TKNTYP' = TOKEN
                        ;       'TKNOFF' = OFFSET PAST COMMAND IN SOURCE STATEMENT.
                        ;
                        ;       JSR     LINSRT
                        ;       BNE     NO ROOM IN MEMORY OR OTHER PROBLEM
                        ;
7ACE  20ED7A            LINSRT  JSR     NUMNAM          ; SETUP 'LINENO' AS STRING NAME.

7AD1  A242                      LDX     #DP-DTAB        ; SETUP STRING AT A POINTER.
7AD3  A000                      LDY     #INLN-DTAB
7AD5  203B9A                    JSR     PMOVE

7AD8  38                        SEC                     ; OFFSET PAST COMMAND =
7AD9  AD6A05                    LDA     TKNOFF          ; ... 'TKNOFF'.
7ADC  E582                      SBC     INLN+2          ; ... - 'INLN+2'.
7ADE  18                        CLC
7ADF  6906                      ADC     #6              ; ... + 6.
7AE1  8D6A05                    STA     TKNOFF

7AE4  4C0599                    JMP     SINSRT          ; INSERT LINE & RETURN WITH CC SET.



7AE7                            PROC
                        ;
                        ; LDELET -- NUMBERED LINE DELETE FROM STATEMENT LIST
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       'LINENO' = LINE # (BINARY)
                        ;
                        ;       JSR     LDELET
                        ;       BNE     LINE NOT FOUND OR OTHER PROBLEM
                        ;
7AE7  20ED7A            LDELET  JSR     NUMNAM          ; SETUP 'LINENO' AS STRING NAME.
7AEA  4CEC98                    JMP     SDELET          ; DELETE LINE & RETURN WITH CC SET.



7AED                            PROC
                        ;
                        ; NUMNAM -- SETUP 'LINENO' AS STRING NAME & SETUP ACCESS TO STATEMENT LIST.
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       JSR     NUMNAM
                        ;
                        ;       'ATRTYP' SET FOR LINE #
                        ;

7AED  A9DC              NUMNAM  LDA     # LOW LINENO
7AEF  85BE                      STA     NP
7AF1  A900                      LDA     # HIGH LINENO
7AF3  85BF                      STA     NP+1
7AF5  A900                      LDA     #0
7AF7  85C0                      STA     NP+2
7AF9  A902                      LDA     #2
7AFB  85C1                      STA     NP+3
7AFD  4C9F9E                    JMP     STMLST          ; SETUP TO ACCESS STATEMENT LIST & RETURN.
```

```
7800                         PROC
                     ;
                     ; GETCOM -- GET A COMMAND LINE FOR THE MAIN LOOP
                     ;
                     ; CALLING SEQUENCE:
                     ;
                     ;       'LOADFG' = 0 IF NOT LOADING FROM DEVICE, ELSE LOADING.
                     ;       'RUN' = 0 IF IMMEDIATE MODE, ELSE RUN MODE.
                     ;       'NXTLN' POINTS TO NEXT RUN MODE LINE.
                     ;
                     ;       JSR     GETCOM
                     ;       BNE     ERROR (A = ERROR NUMBER)
                     ;
                     ;       'INLN' POINTS TO NEW COMMAND LINE.
                     ;       Y = INDEX TO START OF STATEMENT.
                     ;       'NXTLN' POINTS TO NEXT RUN MODE LINE.
                     ;
                     ;       IF 'RUN', THEN
                     ;
                     ;       'TKNTYP' = TOKENIZED COMMAND
                     ;       'TKNOFF' = OFFSET PAST COMMAND IN STATEMENT STORAGE.
                     ;
7800  AD3205         GETCOM  LDA     LOADFG          ; LOADING FROM DEVICE?
7803  D041 ^7846             BNE     :GC200          ; YES.

7805  A5FF                   LDA     RUN             ; RUN MODE?
7807  F031 ^783A             BEQ     :GC100          ; NO -- IMMEDIATE.

7809  207E9F         :GC010  JSR     ABRTCK          ; YES -- CHECK FOR OPERATOR ABORT.

780C  A200                   LDX     #INLN-DTAB      ; GET NEXT STATEMENT ADDRESS.
780E  A004                   LDY     #NXTLN-DTAB
7810  20459A                 JSR     DMOVI

7813  A000                   LDY     #0              ; GET & SAVE LINE END INDEX.
7815  B180                   LDA     (INLN),Y
7817  8583                   STA     INLN+3

7819  A030                   LDY     #81H-DTAB       ; END OF PROGRAM?
781B  20159C                 JSR     DCMPI
781E  D006 ^7826             BNE     :GC020          ; NO -- KEEP TRUCKIN'.

7820  A981                   LDA     #EOFERR         ; RETURN WITH INDICATOR.
7822  8D4305                 STA     NOCCNT          ; NO CONTINUATION.
7825  60                     RTS

7826                 :GC020
                     ; *S*   LDX     #INLN-DTAB
7826  20869A                 JSR     SATTR           ; 'ATTRIBUTE'
7829  8D6805                 STA     TKNTYP          ; AS ADVERTISED.
782C  C8                     INY
782D  B1A1                   LDA     (TEMP),Y
782F  8D6A05                 STA     TKNOFF          ; AS ADVERTISED.

7832  A204                   LDX     #NXTLN-DTAB     ; POINT TO NEXT LINE.
7834  20AA9A                 JSR     SNXTI
7837  A900                   LDA     #0              ; SET CC FOR RETURN.
```

```
7B39  60                  RTS

                  ; GET A LINE FROM THE CONSOLE.

7B3A              :GC100
7B3A  A900                LDA     #0                 ; CLEAR LINE LENGTH FOR "BREAK".
7B3C  8583                STA     INLN+3

7B3E  A200                LDX     #INLN-DTAB         ; GET AN INPUT LINE FROM CONSOLE.
7B40  208194              JSR     GETLIN
7B43  A000                LDY     #0                 ; SET INDEX TO START OF STATEMENT (CC TOO).
7B45  60                  RTS

                  ; GET DATA FROM DEVICE ASSIGNED TO IOCB 3.

7B46  86A1        :GC200  STX     TEMP               ; SAVE REGISTERS.

7B48  A580        :GC205  LDA     INLN               ; SETUP BUFFER ADDRESS.
7B4A  8D7403              STA     IOCB3+ICBAL
7B4D  A581                LDA     INLN+1
7B4F  8D7503              STA     IOCB3+ICBAH

7B52  A905                LDA     #GETR              ; GET RECORD COMMAND.
7B54  8D7203              STA     IOCB3+ICCOM

7B57  A979                LDA     # LOW LINLNG-1     ; SETUP MAXIMUM LINE LENGTH.
7B59  8D7803              STA     IOCB3+ICBLL
7B5C  A9FF                LDA     # HIGH LINLNG-1
7B5E  8D7903              STA     IOCB3+ICBLH

7B61  A230                LDX     #IOCB3             ; GET RECORD.
7B63  2056E4              JSR     CIO

7B66  AD7803              LDA     IOCB3+ICBLL        ; PUT START/END INDICES IN POINTER.
7B69  8583                STA     INLN+3
7B6B  A900                LDA     #0
7B6D  8582                STA     INLN+2

7B6F  C000                CPY     #0                 ; ERROR?
7B71  1029 ^7B9C          BPL     :GC250             ; NO.

7B73  A900                LDA     #0                 ; THAT OR END-OF-FILE.
7B75  8D3205              STA     LOADFG             ; STOP LOADING IN EITHER CASE.

7B78  C088                CPY     #$88               ; END OF FILE?
7B7A  D01D ^7B99          BNE     :GC220             ; NO.

7B7C  203F97              JSR     DCLOSE             ; YES -- CLOSE DEVICE.

7B7F  A5FF                LDA     RUN                ; IS THE USER PROGRAM RUNNING?
7B81  F010 ^7B93          BEQ     :GC210             ; NO -- IMMEDIATE LOAD OR LOAD ERROR.

7B83  A900                LDA     #0                 ; CONTINUE O.K.
7B85  8D0305              STA     NOCONT

7B88  A5AE                LDA     SIL                ; SETUP TO RUN PROGRAM LOADED.
7B8A  8584                STA     RSTLN
```

```
788A  8584                  STA     NXTLN
```

```
788C  A5AF                  LDA     S1L+1
788E  8585                  STA     NXTLN+1
7890  4C0976                JMP     :GC010          ; (TOO FAR FOR 'RELATIVE').

7893  202CB5       :GC210   JSR     RDYMES          ; GENERATE "READY" MESSAGE.
7896  4CE478                JMP     MLRES           ; GRACEFUL TERMINATION OF LOAD.

7899  4C2697       :GC220   JMP     DOP005          ; ABORT LOAD OPERATION.

789C  A000         :GC250   LDY     #0              ; ACCEPT ONLY NUMBERED LINES.
789E  20139F                JSR     SLB
78A1  20839E                JSR     CNUMBR
78A4  B0A2 ^7848            BCS     :GC205          ; NOT NUMBERED--IGNORE.

78A6  A6A1                  LDX     TEMP            ; RESTORE REGISTER.
78A8  A000                  LDY     #0              ; SETUP INDEX TO START OF STATEMENT (=0).
78AA  60                    RTS                     ; RETURN WITH CC SET.
```

```
7BAB                      PROC
                    ;
                    ; SYCMND -- SYNTAX CHECK THE COMMAND
                    ;
                    ; CALLING SEQUENCE:
                    ;
                    ;       X = VALID COMMAND MODE.
                    ;       'INLN' POINTS TO THE STATEMENT
                    ;       Y = INDEX TO START OF STATEMENT
                    ;
                    ;       JSR     SYCMND
                    ;       BNE     SYNTAX ERROR (A = ERROR CODE)
                    ;
                    ;       'TKNTYP' = TOKENIZED COMMAND.
                    ;          0/#USROFF-1                 = COMMAND IS IN 'CDTAB'
                    ;          #USROFF/#USROFF+#USRMAX-1 = COMMAND IS IN 'USRTAB'
                    ;                          'TKNCNT'   = RESERVED FOR ': CONTINUATION'
                    ;                          'TKNNUL'   = RESERVED FOR 'NULL' COMMAND
                    ;       'TKNOFF' = OFFSET PAST COMMAND FROM BEGINNING OF STATEMENT.
                    ;
7BAB  20D39E        SYCMND  JSR     SCNLBL          ; SCAN PAST LABEL IF PRESENT.
7BAE  20139F                JSR     SLB

7BB1  C93A                  CMP     #':'            ; COMMAND CONTINUATION?
7BB3  F01D ^7BD2            BEQ     :SC010          ; YES.

7BB5  20F99E                JSR     CHKTRM          ; 'NULL' COMMAND?
7BB8  F01D ^7BD7            BEQ     :SC020          ; YES.

7BBA  A900                  LDA     #0              ; RESET EXECUTE FLAG.
7BBC  8592                  STA     EXEC

7BBE  20567C                JSR     CMATCH          ; FIND COMMAND.
7BC1  F006 ^7BC9            BEQ     :SC005          ; VALID.

7BC3  C902                  CMP     #IVCERR         ; IF NOT IN TABLE, ASSUME 'GR:'.
7BC5  D01A ^7BE1            BNE     :SC099          ; ERROR.

7BC7  A23A                  LDX     #CDG-CDTAB      ; CASE: 10 360(HOME;DRAW 10;TURN 1).

7BC9  8E6805        :SC005  STX     TKNTYP          ; TOKENIZE COMMAND.
7BCC  8C6A05                STY     TKNOFF          ; OFFSET PAST COMMAND.
7BCF  4C087C                JMP     EXC100

7BD2  C8            :SC010  INY                     ; MOVE PAST ':'.
7BD3  A9FE                  LDA     #TKNCNT         ; COMMAND CONTINUATION.
7BD5  D002 ^7BD9            BNE     :SC050          ; (BRA).

7BD7  A9FF          :SC020  LDA     #TKNNUL         ; NULL COMMAND.

7BD9  8D6805        :SC050  STA     TKNTYP
7BDC  8C6A05                STY     TKNOFF

7BDF  A900                  LDA     #0              ; SET CC FOR EXIT.

7BE1  60            :SC099  RTS
```

```
7BE2                        PROC
                    ;
                    ; EXCMND -- EXECUTE THE COMMAND
                    ;
                    ; CALLING SEQUENCE:
                    ;
                    ;       'TKNTYP' = TOKENIZED COMMAND.
                    ;       'TKNOFF' = OFFSET PAST COMMAND.
                    ;
                    ;       JSR     EXCMND
                    ;       BNE     SYNTAX OR RUN-TIME ERROR (A = ERROR CODE).
                    ;
7BE2  AC6A05        EXCMND  LDY     TKNOFF          ; OFFSET PAST COMMAND.
7BE5  A9FF                  LDA     #$FF            ; SET EXECUTE FLAG.
7BE7  8592                  STA     EXEC

7BE9  AE6805                LDX     TKNTYP          ; TRAP FOR 'RESERVED' TOKENS.
7BEC  E0FE                  CPX     #TKNCNT
7BEE  9018 ^7C08            BCC     EXC100          ; NOT 'RESERVED'.
7BF0  D013 ^7C05            BNE     :EC020          ; 'NULL' COMMAND.

                    ; COMMAND CONTINUATION

7BF2  AE6905                LDX     LSTKN           ; USE TOKEN FROM 'LAST' COMMAND.
7BF5  8E6805                STX     TKNTYP

7BF8  E008                  CPX     #CLNCNT         ; NO -- CHECK 'CDTAB' SEGMENT.
                                                    ; ('USRTAB' NOT ALLOWED).
7BFA  9004 ^7C00            BCC     :EC010          ; O.K.

7BFC  88                    DEY                     ; POINT TO ':'.
7BFD  A902                  LDA     #IVCERR         ; INVALID CONTINUATION.
7BFF  60                    RTS

                    ; COMMAND CONTINUATION IS VALID.

7C00  AD0605        :EC010  LDA     EXECF           ; USE PRIOR 'EXECF'.
7C03  D03F ^7C44            BNE     :EC500          ; EXECUTE COMMAND USING PRIOR 'XJUMP'.

                    ; EXIT FOR 'NULL' COMMAND.

7C05  A900          :EC020  LDA     #0              ; SET CC FOR EXIT.
7C07  60                    RTS

                    ; *** ENTRY FROM 'SYCMND' ***

7C08  200E81        EXC100  JSR     COND            ; PROCESS CONDITION IF PRESENT.
7C0B  A592                  LDA     EXEC            ; EXECUTE MODE?
7C0D  F005 ^7C14            BEQ     :EC300          ; NO -- SYNTAX SCAN ONLY.

7C0F  AD0605                LDA     EXECF           ; EXECUTE COMMAND?
7C12  F041 ^7C55            BEQ     :EC900          ; NO -- NORMAL EXIT.

7C14  84AB          :EC300  STY     XTEMP           ; SAVE Y.
7C16  AC6805                LDY     TKNTYP          ; 'USRTAB' OR 'CDTAB'?
7C19  8C6905                STY     LSTKN           ; SAVE TOKEN IN CASE NEXT COMMAND USES
```

```
                                                   ; ':-CONTINUATION'.
7C1C  C06C                  CPY     #USROFF
7C1E  900F ^7C2F            BCC     :EC400         ; 'CDTAB'.

7C20  98                    TYA                    ; 'NORMALIZE' RELATIVE TO 'USRTAB'.
7C21  38                    SEC
7C22  E96C                  SBC     #USROFF
7C24  A8                    TAY
7C25  AD0005                LDA     USRTAB
7C28  85A1                  STA     TEMP
7C2A  AD0105                LDA     USRTAB+1
7C2D  D006 ^7C35            BNE     :EC410         ; (BRA).

7C2F  A93E         :EC400   LDA     # LOW CDTAB
7C31  85A1                  STA     TEMP
7C33  A980                  LDA     # HIGH CDTAB

7C35  85A2         :EC410   STA     TEMP+1
7C37  B1A1                  LDA     (TEMP),Y       ; MOVE ADDRESS TO JUMP INSTRUCTION.
7C39  8D0805                STA     XJUMP+1
7C3C  C8                    INY
7C3D  B1A1                  LDA     (TEMP),Y
7C3F  8D0905                STA     XJUMP+2

7C42  A4A8                  LDY     XTEMP          ; RESTORE INDEX.

7C44  A592         :EC500   LDA     EXEC           ; SET CC FOR X-ROUTINES.
7C46  200705                JSR     XJUMP          ; YES -- EXECUTE (OR SCAN).
7C49  D00A ^7C55            BNE     :EC900         ; ERROR -- RETURN WITH CC SET.

7C4B  20139F                JSR     SLB            ; SKIP ANY BLANKS.
7C4E  20F99E                JSR     CHKTRM         ; STATEMENT TERMINATOR?
7C51  F002 ^7C55            BEQ     :EC900         ; YES -- O.K.

7C53  A902                  LDA     #JNKERR        ; JUNK -- ERROR.

7C55  60           :EC900   RTS                    ; RETURN WITH CC SET.
```

```
7C56                       PROC
                     ;
                     ; CMATCH -- COMMAND MATCH ROUTINE
                     ;
                     ; ORDER OF SEARCHING:
                     ;
                     ;       1. THE USER EXTENDABLE COMMAND TABLE
                     ;       2. THE GRAPHICS SUBCOMMANDS
                     ;       3. THE INTERNAL COMMAND TABLE
                     ;
                     ;
                     ; CALLING SEQUENCE:
                     ;
                     ;       'USRTAB' = ADDRESS OF USER EXTENDABLE COMMAND TABLE (0=NONE).
                     ;                  (OFFSETS ARE RELATIVE TO 'USRTAB').

                             X = IMMEDIATE AND/OR RUN COMMAND' VALID
                     ;       'INLN' POINTS TO SOURCE STATEMENT.
                     ;       Y = INDEX TO START OF COMMAND NAME.
                     ;
                     ;       JSR     CMATCH
                     ;       BNE     NO MATCH IN TABLE (A = ERROR CODE, Y UNCHANGED)
                     ;
                     ;       X =  VALUE OF 'CTAB' DATA BYTE FOR ENTRY ('TOKENIZED' COMMAND).
                     ;       X <  'USROFF' (OFFSET IN 'CDTAB').
                     ;       X >= 'USROFF' ('USROFF' + OFFSET IN 'USRTAB').
                     ;       Y =  INDEX TO START OF FIELD AFTER COMMAND NAME.
                     ;       'CTABAT' = ATTRIBUTE BITS OF COMMAND.
                     ;
                     ; NOTE: NAME MATCH MUST BE EXACT FOR THE REST OF THE
                     ;       STATEMENT TO BE PROCESSED CORRECTLY. FOR EXAMPLE:
                     ;               "TYPEN:" WILL BE SCANNED AS TY<JUNK>:, NOT
                     ;                                           T<JUNK>N:.
                     ;
7C56  8E1005         CMATCH  STX     CTABAT          ; SAVE VALID COMMAND TYPES.
7C59  84A6                   STY     XTEMP           ; SAVE Y REG.
7C5B  AD0005                 LDA     USRTAB          ; SELECT 'USRTAB' IF ADDR>255.
7C5E  8590                   STA     TABADR
7C60  AD0105                 LDA     USRTAB+1
7C63  F018 ^7C7D             BEQ     :CMA10          ; NO USER EXTENDED COMMAND TABLE.

7C65  8591                   STA     TABADR+1
7C67  20BD7C                 JSR     CMACOM          ; SEARCH 'USRTAB'
7C6A  D011 ^7C7D             BNE     :CMA10          ; NOT IN 'USRTAB'

7C6C  E092                   CPX     #USRMAX         ; IS 'USRTAB' TOO LARGE?
7C6E  B00D ^7C7D             BCS     :CMA10          ; YES -- PRETEND COMMAND WAS NOT THERE.

7C70  98                     TYA                     ; CHECK 'ATTRIBUTE'
7C71  2C1005                 BIT     CTABAT
7C74  F031 ^7CA7             BEQ     :CMA90          ; WRONG COMMAND TYPE.

                     ; COMMAND IS IN 'USRTAB'

7C76  8A                     TXA                     ; SET 'TOKEN' FOR 'USRTAB'
7C77  18                     CLC
7C78  696C                   ADC     #USROFF         ; SET OFFSET TO IDENTIFY TOKEN IN 'USRTAB'.
```

```
7C7A  AA                      TAX
7C7B  D023 ^7CA0              BNE     :CMA50            ; (BRA).

                        ; SEARCH GRAPHICS SUBCOMMANDS

7C7D  A4AB          :CMA10    LDY     XTEMP             ; RESTORE INDEX.
7C7F  A208                    LDX     #GTABX
7C81  20A87C                  JSR     SBCMAT            ; GRAPHICS SUBCOMMAND?
7C84  D007 ^7C8D              BNE     :CMA20            ; NO.

7C86  A4AB                    LDY     XTEMP             ; RESTORE INDEX FOR SYNTAX CHECK.
7C88  A23B                    LDX     #CDG-CDTAB        ; TOKENIZE AS 'GR:'.
7C8A  A900                    LDA     #0                ; SET CC FOR EXIT.
7C8C  60                      RTS

                        ; SEARCH INTERNAL COMMAND TABLE

7C8D  A901          :CMA20    LDA     #LOW CTAB
7C8F  8590                    STA     TABADR
7C91  A97C                    LDA     #HIGH CTAB
7C93  8591                    STA     TABADR+1
7C95  20B07C                  JSR     CMACOM            ; SEARCH 'CTAB'
7C98  D01E ^7CB8              BNE     :CMA99            ; NOT IN 'CTAB' -- INVALID.

7C9A  98                      TYA
7C9B  2C1005                  BIT     CTABAT
7C9E  F007 ^7CA7              BEQ     :CMA90            ; WRONG COMMAND TYPE.

                        ; COMMAND IS IN 'CTAB'

7CA0  8C1005        :CMA50    STY     CTABAT            ; STORE 'ATTRIBUTE'.
7CA3  A900                    LDA     #0                ; SET CC FOR EXIT.
7CA5  F011 ^7CB8              BEQ     :CMA99            ; (BRA).

7CA7  A983          :CMA90    LDA     #NRCERR           ; WRONG COMMAND TYPE.
7CA9  D00D ^7CB8              BNE     :CMA99            ; (BRA).


                        ;
                        ; SBCMAT -- SUBCOMMAND MATCH ROUTINE
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       X = INDEX TO THE SUBCOMMAND TABLE FROM 'SBCTAB'
                        ;       'INLN' POINTS TO SOURCE STATEMENT.
                        ;       Y = INDEX TO START OF SUBCOMMAND NAME.
                        ;
                        ;       JSR SBCMAT
                        ;       BNE NOMATCH IN TABLE (A = ERROR CODE, Y UNCHANGED)
                        ;
                        ;       X = VALUE OF 'SBCTAB' DATA BYTE FOR ENTRY ('OFFSET' OR 'VALUE')
                        ;       Y = INDEX TO START OF FIELD AFTER COMMAND NAME.
                        ;
7CA8  BD207E        SBCMAT    LDA     SBCTAB,X          ; SELECT SUBCOMMAND TABLE.
7CAB  8590                    STA     TABADR
7CAD  BD217E                  LDA     SBCTAB+1,X
```

```
7CB3  8591                STA     TABADR+1
7CB5  20BD7C              JSR     CMACON            ; COMMON CODE.

                  ; *** OPTIONAL ENTRY FROM 'CMATCH' ***


7CB8              :CMA99
7CB8  08                  PHP                       ; SAVE CC.
7CB9  A4A1                LDY     TEMP              ; RESTORE INDEX IN 'INLN'.
7CBB  28                  PLP                       ; RESTORE CC FOR CALLER.
7CBC  60                  RTS


7CBD                      PROC
                  ;
                  ; CMACON -- COMMON CODE FOR 'CMATCH' AND 'SBCMAT'
                  ;
                  ; CALLING SEQUENCE:
                  ;
                  ;       'TABADR' = BASE ADDRESS OF MATCH TABLE.
                  ;       'INLN' POINTS TO SOURCE STATEMENT.
                  ;       Y = CURRENT INDEX IN 'INLN'
                  ;
                  ;       JSR CMACON
                  ;       BNE NO MATCH IN TABLE (A = ERROR CODE)
                  ;
                  ;       X = 'OFFSET' BYTE
                  ;       Y = 'ATTRIBUTE' BYTE
                  ;       TEMP = INDEX TO START OF FIELD AFTER COMMAND NAME.
                  ;
                  ;
                  ; ALAS THE INDIRECT INDEXING OF THE 6502.
                  ;       MOVE '(INLN),Y' THROUGH '(INLN),Y + INBFSZ-1 TO
                  ;       A FIXED BUFFER, 'INLNBF', SO THAT 'X' CAN INDEX 'INLN'
                  ;       WHILE 'Y' INDEXES THE TABLE.
                  ;
                  ;       CONVERT LC -> UC IN 'INLNBF'.
7CBD  20139F      CMACON  JSR     SLB               ; SKIP LOADING BLANKS
7CC0  84A1                STY     TEMP              ; SAVE INDEX IN 'INLN'
7CC2  20F79B              JSR     MVINLN            ; MOVE 'PART' OF 'INLN'

7CC5  A0FF                LDY     #$FF              ; SEARCH FROM THE BEGINNING OF '(TABADR)'
                                                    ; (PRE-DECREMENT).
7CC7  A2FF                LDX     #$FF              ; START AT TWO BEGINNING OF THE SOURCE.
                                                    ; (PRE-DECREMENT).

7CC9  C8          :CMA05  INY                       ; NEXT TABLE CHARACTER.
7CCA  E8                  INX                       ; NEXT SOURCE CHARACTER.
7CCB  B190                LDA     (TBLBAS),Y        ; SEE IF END OF NAME IN TABLE.

7CCD  3022 ^7CF1  :CMA10  BMI     :CMA70            ; YES -- MATCH FOUND.

7CCF  DD3805              CMP     INLNBF,X          ; MATCH NEXT SOURCE CHAR?
7CD2  F0F5 ^7CC9          BEQ     :CMA05            ; YES -- CONTINUE COMPARISON.
```

```
7CD4  C8            :CMA20  INY                         ; SCAN TO END OF NAME ENTRY.
7CD5  B190                  LDA    (TBLBAS),Y
7CD7  10FB ^7CD4            BPL    :CMA20

7CD9  C8                    INY                         ; SCAN PAST 'ATTRIBUTE' BYTE.
7CDA  C8                    INY                         ; SCAN PAST 'OFFSET' BYTE.

7CDB  C0F2                  CPY    #$FF-INBFSZ-3        ; WILL INDEX WRAP?
7CDD  9008 ^7CE7            BCC    :CMA30               ; NO.
7CDF  98                    TYA                         ; YES -- ADJUST BASE POINTER.
7CE0  A210                  LDX    #TBLBAS-DTAB
7CE2  20089D                JSR    DADDP
7CE5  A000                  LDY    #0                   ; ... AND RESET INDEX.

7CE7  A200          :CMA30  LDX    #0                   ; RESTORE SOURCE INDEX.
7CE9  B190                  LDA    (TBLBAS),Y           ; CHECK FOR END OF TABLE.
7CEB  D0E0 ^7CCD            BNE    :CMA10               ; NO -- KEEP CHECKING.

7CED  A902                  LDA    #IVCERR              ; TABLE END -- INVALID COMMAND.
7CEF  D00F ^7D00            BNE    :CMA90               ; (BRA).

7CF1  48            :CMA70  PHA                         ; VALUE OF 'ATTRIBUTE' BYTE.
7CF2  8A                    TXA                         ; OFFSET IN 'INLNBF'.
7CF3  18                    CLC
7CF4  65A1                  ADC    TEMP                 ; + INITIAL OFFSET IN 'INLN'.
7CF6  85A1                  STA    TEMP                 ; INDEX TO START OF FIELD AFTER NAME.

7CF8  C8                    INY
7CF9  B190                  LDA    (TBLBAS),Y           ; 'OFFSET' BYTE.
7CFB  AA                    TAX

7CFC  68                    PLA                         ; VALUE OF 'ATTRIBUTE' BYTE.
7CFD  A8                    TAY

7CFE  A900                  LDA    #0                   ; SET CC FOR EXIT.

7D00  60            :CMA90  RTS
```

```
7D01                           PROC
                          ;
                          ; 'USRTAB' -- USER EXTENDABLE COMMAND TABLE
                          ;
                          ; SAME STRUCTURE AS 'CTAB'
                          ;
                          ; THE EQUIVALENT 'CDTAB' IS APPENDED TO 'USRTAB' SO THAT
                          ; THE OFFSETS ARE ACTUALLY FROM THE BEGINNING OF 'USRTAB'.
                          ;
                          ; THE TOTAL LENGTH OF 'USRTAB' MAY NOT EXCEED 'USRMAX'.
                          ;
                          ; COMMAND TABLE
                          ;
                          ; CONSISTS OF N ENTRIES, EACH OF THE FOLLOWING FORMAT:
                          ;
                          ;       DB  "<COMMAND NAME>",
                          ;           $80+[IMMEDIATE]+[RUN]+[: REQUIRED],
                          ;           INDEX TO COMMAND DATA TABLE.
                          ;
                          ; THE TABLE IS ENDED BY "<NAME>" = 0.
                          ; ORDER OF ENTRIES IS ONLY RESTRICTED BY FIRST FOUND - FIRST MATCHED, NOT BEST FIT.
                          ;

      = 0080         SB       = $80                     ; SIGN BIT.
      = 0040         CTIMM    = $40                     ; IMMEDIATE COMMAND.
      = 0020         CTRUN    = $20                     ; RUN COMMAND.
      = 0010         CTCLN    = $10                     ; : REQUIRED.
      = 0060         CTBOTH   = CTIMM+CTRUN             ; IMMEDIATE OR RUN COMMAND.
      = 0060         CTNORM   = CTBOTH+CTCLN-CTCLN      ; IMMEDIATE OR RUN COMMAND, : REQUIRED.

      = 7D01         CTAB     = *                       ; INTERNAL COMMAND TABLE BASE ADDRESS.

7D01  44454C4159              DB    'DELAY',SB+CTNORM,:CDSPD-CDTAB ; DELAY.

7D08  4C495354C0              DB    'LIST',SB+CTIMM,:CDLST-CDTAB  ; LIST STORED PROGRAM.

7D0E  44454CC00A              DB    'DEL',SB+CTIMM,:CDDEL-CDTAB   ; DELETE RANGE OF LINES.

7D13  52554EC00E              DB    'RUN',SB+CTIMM,:CDRUN-CDTAB   ; RUN STORED PROGRAM.

      = 0000                  IF    DOS
  -                           DB    'DOS',SB+CTIMM,:CDDOS-CDTAB   ; GO TO DOS UTILITY.
                              ENDIF

7D18  53415645C0              DB    'SAVE',SB+CTIMM,:CDSAV-CDTAB  ; SAVE STORED PROGRAM.

7D1E  4E4557C048              DB    'NEW',SB+CTIMM,:CDNEW-CDTAB   ; CLEAR PROGRAM & VARS.

7D23  4155544FC0              DB    'AUTO',SB+CTIMM,:CDAUT-CDTAB  ; AUTO-INPUT.

7D29  52454EC01A              DB    'REN',SB+CTIMM,:CDREN-CDTAB   ; RENUMBER PROGRAM.

7D2E  434F4E54C0              DB    'CONT',SB+CTIMM,:CDCON-CDTAB  ; CONTINUE.

7D34  5053C01C                DB    'PS',SB+CTIMM,:CDCOL-CDTAB    ; TURTLE PEN STATUS.
```

```
7D38  4553C01E            DB   'ES',SB+CTIMM,:CDENS-CDTAB    ; TURTLE ENVIRONMENT STATUS.
7D3C  434F4C4F52          DB   'COLORS',SB+CTIMM,:CDPAL-CDTAB ; PALETTE OF COLORS.
7D44  444952C022          DB   'DIR',SB+CTIMM,:CDDIR-CDTAB   ; DISK DIRECTORY.
7D49  48454C50C0          DB   'HELP',SB+CTIMM,:CDCOM-CDTAB  ; COMMAND LIST.

7D4F  44554D50E0          DB   'DUMP',SB+CTBOTH,:CDDMP-CDTAB ; DUMP.
7D55  4C4F4144E0          DB   'LOAD',SB+CTBOTH,:CDLOD-CDTAB ; LOAD.
7D5B  4D45524745          DB   'MERGE',SB+CTIMM,:CDMRG-CDTAB ; MERGE.
7D62  415050454E          DB   'APPEND',SB+CTIMM,:CDAPP-CDTAB ; APPEND.
7D6A  5452414345          DB   'TRACE',SB+CTBOTH,:CDTRC-CDTAB ; TRACE.
7D71  564E4557E0          DB   'VNEW',SB+CTBOTH,:CDNWV-CDTAB  ; VNEW.
7D77  5456E04C            DB   'TV',SB+CTBOTH,:CDTV-CDTAB     ; TV.
7D7B  43414C4CE0          DB   'CALL',SB+CTNORM,:CDCAL-CDTAB  ; CALL.
7D81  54415045E0          DB   'TAPE',SB+CTNORM,:CDCSS-CDTAB  ; CASSETTE ON/OFF.
7D87  5453594E43          DB   'TSYNC',SB+CTNORM,:CDSNC-CDTAB ; CASSETTE SYNC.
7D8E  52454144E0          DB   'READ',SB+CTNORM,:CDIN-CDTAB   ; READ RECORD.
7D94  5752495445          DB   'WRITE',SB+CTNORM,:CDOUT-CDTAB ; WRITE RECORD.
7D9B  434C4F5345          DB   'CLOSE',SB+CTNORM,:CDDON-CDTAB ; CLOSE FILE.
7DA2  54E000              DB   'T',SB+CTNORM,:CDT-CDTAB       ; TYPE.
7DA5  4C45545445          DB   'LETTERS',SB+CTRUN,:CDLTR-CDTAB ; LETTERS
7DAE  414BE064            DB   'AK',SB+CTNORM,:CDAK-CDTAB     ; ACCEPT KEYSTROKE.
7DB2  4158E062            DB   'AX',SB+CTNORM,:CDAX-CDTAB     ; ACCEPT LITERAL.
7DB6  41E02E              DB   'A',SB+CTNORM,:CDA-CDTAB       ; ACCEPT.
7DB9  43E030              DB   'C',SB+CTNORM,:CDC-CDTAB       ; COMPUTE.
7DBC  55E032              DB   'U',SB+CTNORM,:CDU-CDTAB       ; USE.
7DBF  45E034              DB   'E',SB+CTNORM,:CDE-CDTAB       ; END.
7DC2  4A4DE036            DB   'JM',SB+CTNORM,:CDJM-CDTAB     ; JUMP ON MATCH.
7DC6  4AE038              DB   'J',SB+CTNORM,:CDJ-CDTAB       ; JUMP.
7DC9  4752E03A            DB   'GR',SB+CTNORM,CDG-CDTAB       ; GRAPHICS.
```

```
7DCD  4D5358E03C          DB   'MSX',SB+CTNORM,:CDMSX-CDTAB
7DD2  4D53E03E            DB   'MS',SB+CTNORM,:CDMS-CDTAB     ; MATCH (PRODUCING) STRING.
7DD6  4D58F040            DB   'MX',SB+CTNORM,:CDMX-CDTAB
7DDA  4DE042              DB   'M',SB+CTNORM,:CDM-CDTAB       ; MATCH.
7DDD  534FE044            DB   'SO',SB+CTNORM,:CDS-CDTAB      ; SOUNDS.
7DE1  52E006              DB   'R',SB+CTNORM,:CDR-CDTAB       ; REMARK.
7DE4  5041E046            DB   'PA',SB+CTNORM,:CDW-CDTAB      ; PAUSE.
7DE8  59E002              DB   'Y',SB+CTNORM,:CDY-CDTAB       ; TYPE IF YES.
7DEB  4EE004              DB   'N',SB+CTNORM,:CDN-CDTAB       ; TYPE IF NO.
7DEE  504F53E054          DB   'POS',SB+CTNORM,:CDPOS-CDTAB  ; POSITION.
7DF3  53544F50A0          DB   'STOP',SB+CTRUN,:CDSTP-CDTAB  ; STOP.
7DF9  5345545045          DB   'SETPEN',SB+CTNORM,:CDSTC-CDTAB ; SET COLOR.
7E01  5345544C45          DB   'SETLET',SB+CTNORM,:CDSTL-CDTAB    ; SET LETTERS COLOR.
7E09  5343524F4C          DB   'SCROLL',SB+CTBOTH,:CDSCR-CDTAB ; SCROLL SELECT.
7E11  5353415645          DB   'SSAVE',SB+CTNORM,:CDSSA-CDTAB  ; SCREEN SAVE.
7E18  534C4F4144          DB   'SLOAD',SB+CTNORM,:CDSLO-CDTAB  ; SCREEN LOAD.
7E1F  00                  DB   0                               ; END OF TABLE.
```

```
                        ;
                        ; SUBCOMMAND TABLES
                        ;
                        ; THERE CAN BE UP TO 128 SUBCOMMAND TABLES.
                        ; THE STRUCTURE OF EACH IS IDENTICAL TO THE COMMAND TABLE EXCEPT:
                        ;       THE 'OFFSET' BYTE CAN REPRESENT A 'VALUE', WITH THE
                        ;       CALLER DECIDING WHICH.
                        ;
                        ; THE CALLER SELECTS WHICH SUBCOMMAND TABLE BY SETTING ON INDEX
                        ; TO THE TABLE ADDRESS FROM 'SBCTAB'.
                        ;

      = 7E20            SBCTAB  = *                     ; BASE ADDRESS OF SUBCOMMAND TABLE ADDRESSES.
      = 0000            OPTABX  = *-SBCTAB              ; NUMERICAL/RELATONAL OPERATIONS (BINARY).
 7E20   347E                    DW OPTAB
      = 0002            UNTABX  = *-SBCTAB              ; UNARY OPERATORS.
 7E22   677E                    DW UNTAB
      = 0004            GTABX   = *-SBCTAB              ; GRAPHICS SUBCOMMAND TABLE.
 7E24   707E                    DW GTAB
      = 0006            PCTABX  = *-SBCTAB              ; PEN COLOR TABLE.
 7E26   707F                    DW PCTAB
      = 0008            UPDWNX  = *-SBCTAB              ; UP/DOWN TABLE.
 7E28   E17F                    DW UPDTAB
      = 000A            ONOFFX  = *-SBCTAB              ; ON/OFF COMMAND TABLE.
 7E2A   EC7F                    DW ONFTAB
      = 000C            LTTABX  = *-SBCTAB              ; LETTERS COMMAND TABLE.
 7E2C   F67F                    DW LTRTAB
      = 000E            EDTABX  = *-SBCTAB              ; EDGE COMMAND TABLE.
 7E2E   0080                    DW EDGTAB
      = 0010            SCTABX  = *-SBCTAB              ; SCROLL OPTION TABLE.
 7E30   2880                    DW SCRLTB
      = 0012            WLTABX  = *-SBCTAB              ; WALL OPTION TABLE.
 7E32   3780                    DW WALLTB

      = 7E34            OPTAB = *                       ; NUMERIC/RELATIONAL OPERATORS (BINARY).

 7E34   2B8000                  DB   '+',SB,:CDPLS-SBDTAB

 7E37   2D8002                  DB   '-',SB,:CDSUB-SBDTAB

 7E3A   2F8004                  DB   '/',SB,:CDDIV-SBDTAB

 7E3D   2A8006                  DB   '*',SB,:CDMUL-SBDTAB

 7E40   3C3E8008                DB   '<>',SB,:CDNE-SBDTAB

 7E44   3E3D800A                DB   '>=',SB,:CDGE-SBDTAB

 7E48   3C3D800C                DB   '<=',SB,:CDLE-SBDTAB

 7E4C   3D800E                  DB   '=',SB,:CDEQ-SBDTAB

 7E4F   3E8012                  DB   '>',SB,:CDGT-SBDTAB

 7E52   3C8010                  DB   '<',SB,:CDLT-SBDTAB

 7E55   5C8014                  DB   BSLASH,SB,:CDMOD-SBDTAB
```

```
7E55  5C8014            DB   BSLASH,SB,:CDMOD-SBDTAB
```

```
7E58  414E448016        DB   'AND',SB,:CDAND-SBDTAB
7E5D  4F528018          DB   'OR',SB,:CDOR-SBDTAB
7E61  584F52801A        DB   'XOR',SB,:CDXOR-SBDTAB
      = 0000            IF   LOGGRP
   -                    DB   'LAND',SB,:CDLAN-SBDTAB
   -                    DB   'LOR',SB,:CDLOR-SBDTAB
                        ENDIF
7E66  00                DB   0                       ; END OF TABLE.
      = 7E67     UNTAB  = *                          ; UNARY OPERATORS.
7E67  2D801C            DB   '-',SB,:CDUMI-SBDTAB
7E6A  4E4F54801E        DB   'NOT',SB,:CDNOT-SBDTAB
7E6F  4C4E4F5480        DB   'LNOT',SB,:CDLNO-SBDTAB
7E75  00                DB   0                       ; END OF TABLE.
      = 7E76     GTAB = *                            ; GRAPHICS SUB-COMMAND TABLE
7E76  4452415754        DB   'DRAWTO',SB,:CDDRT-SBDTAB
7E7E  4452415780        DB   'DRAW',SB,:CDDRW-SBDTAB
7E84  5455524E54        DB   'TURNTO',SB,:CDTNT-SBDTAB
7E8C  5455524E80        DB   'TURN',SB,:CDTRN-SBDTAB
7E92  474F544F80        DB   'GOTO',SB,:CDGOT-SBDTAB
7E98  46494C4C54        DB   'FILLTO',SB,:CDFIT-SBDTAB
7EA0  46494C4C80        DB   'FILL',SB,:CDFIL-SBDTAB
7EA6  474F802C          DB   'GO',SB,:CDGO-SBDTAB
7EAA  4348414E47        DB   'CHANGE',SB,:CDCHG-SBDTAB
7EB2  50454E8032        DB   'PEN',SB,:CDPEN-SBDTAB
7EB7  434C454152        DB   'CLEARPENS',SB,:CDCLP-SBDTAB
7EC2  434C454152        DB   'CLEAR',SB,:CDCLR-SBDTAB
7EC9  5155495480        DB   'QUIT',SB,:CDEXI-SBDTAB
7ECF  46554C4C80        DB   'FULL',SB,:CDFUL-SBDTAB
```

```
7ED5  53504C4954         DB    'SPLIT',SB,:CDSPT-SBDTAB
7EDC  57414C4C80         DB    'WALL',SB,:CDWAL-SBDTAB
7EE2  4544474580         DB    'EDGE',SB,:CDEDG-SBDTAB
7EE8  484F4D4580         DB    'HOME',SB,:CDHOM-SBDTAB
7EEE  4E4F525448         DB    'NORTH',SB,:CDNRT-SBDTAB
7EF5  545552544C         DB    'TURTLE',SB,:CDTRT-SBDTAB
7EFD  5348414445         DB    'SHADE',SB,:CDSHD-SBDTAB
7F04  4D4F444580         DB    'MODE',SB,:CDMDE-SBDTAB
7F0A  4241434B47         DB    'BACKGROUND',SB,:CDBCK-SBDTAB
7F16  5345544880         DB    'SETH',SB,:CDTNT-SBDTAB
7F1C  5345544247         DB    'SETBG',SB,:CDBCK-SBDTAB
7F23  434C45414E         DB    'CLEAN',SB,:CDCLR-SBDTAB
7F2A  46448024           DB    'FD',SB,:CDDRW-SBDTAB
7F2E  424B8050           DB    'BK',SB,:CDBK-SBDTAB
7F32  52548028           DB    'RT',SB,:CDTRN-SBDTAB
7F36  4C548052           DB    'LT',SB,:CDLTU-SBDTAB
7F3A  534554504F         DB    'SETPOS',SB,:CDDRT-SBDTAB
7F42  5245504541         DB    'REPEAT',SB,:CDRPT-SBDTAB
7F4A  524F424F54         DB    'ROBOT',SB,:CDRBT-SBDTAB
7F51  4559455380         DB    'EYES',SB,:CDEYS-SBDTAB
7F57  5250454E80         DB    'RPEN',SB,:CDRPN-SBDTAB
7F5D  484F524E80         DB    'HORN',SB,:CDHRN-SBDTAB
7F63  5044805A           DB    'PD',SB,:CDPD-SBDTAB
7F67  50558058           DB    'PU',SB,:CDPU-SBDTAB
7F6B  50458056           DB    'PE',SB,:CDPE-SBDTAB
7F6F  00                 DB    0                      ; END OF TABLE.
      = 7F70      PCTAB = *                           ; PEN COLOR TABLE.
7F70  5245448042         DB    'RED',SB,CRED
7F75  59454C4C4F         DB    'YELLOW',SB,CYELLO
```

```
7F7D  475245454E          DB   'GREEN',SB,$C6
7F84  424C554580          DB   'BLUE',SB,CBLUE
7F8A  424C41434B          DB   'BLACK',SB,CBLACK
7F91  5748495445          DB   'WHITE',SB,$0E
7F98  4F52414E47          DB   'ORANGE',SB,$F4
7FA0  505552504C          DB   'PURPLE',SB,$52
7FA8  4752415980          DB   'GRAY',SB,$04
7FAE  53494C5645          DB   'SILVER',SB,$06
7FB6  474F4C4480          DB   'GOLD',SB,$28
7FBC  50494E4B80          DB   'PINK',SB,$46
7FC2  4C41564544          DB   'LAVENDER',SB,$64
7FCC  42524F574E          DB   'BROWN',SB,$E0
7FD3  4245494745          DB   'BEIGE',SB,$FE
7FDA  4552415345          DB   'ERASE',SB,0
      = 7FE1      UPDTAB = *                       ; UP/DOWN TABLE.
7FE1  55508080    PCTUP   DB   'UP',SB,PCUP
7FE5  444F574E80  PCTDN   DB   'DOWN',SB,PCDN
7FEB  00                  DB   0                   ; END OF TABLE.
      = 7FEC      ONFTAB = *                       ; ON/OFF COMMAND TABLE.
7FEC  4F4E8001            DB   'ON',SB,KON
7FF0  4F46468000          DB   'OFF',SB,KOFF
7FF5  00                  DB   0                   ; END OF TABLE.
      = 7FF6      LTRTAB  = *                      ; LETTERS COMMAND TABLE.
7FF6  534D414C4C          DB   'SMALL',SB,LSMLL
7FFD  4D45444955          DB   'MEDIUM',SB,LMED
8005  4C41524745          DB   'LARGE',SB,LLRG
800C  00                  DB   0                   ; END OF TABLE.
      = 800D      EDGTAB  = *                      ; EDGE COMMAND TABLE.
```

```
8000  5752415080          DB   'WRAP',SB,EWRAP
8013  48414C5480          DB   'HALT',SB,EHALT
8019  424F554E43          DB   'BOUNCE',SB,EBNC
8021  4652454580          DB   'FREE',SB,EFREE
8027  00                  DB   0                   ; END OF TABLE.
      = 8028    SCRLTB    = *                      ; SCROLL OPTION COMMAND.
8028  434F415253          DB   'COARSE',SB,0
8030  46494E4580          DB   'FINE',SB,$FF
8036  00                  DB   0
      = 8037    WALLTB    = *                      ; WALL OPTION
8037  4E4F4E4580          DB   'NONE',SB,0
803D  00                  DB   0                   ; END OF TABLE.
```

```
                    ;
                    ; COMMAND DATA TABLE
                    ;
                    ; CONSISTS OF N WORDS, THE INDICES TO THIS TABLE ARE
                    ; CONTAINED IN 'CTAB'.  THE TOTAL NUMBER OF BYTES IN THE TABLE MAY NOT
                    ; EXCEED 128.
                    ;
                    ; NOTE:  THIS OFFSET IS USED TO 'TOKENIZE' THE COMMAND.
                    ;        THE 'MSB' FLAGS THAT THE COMMAND IS IN 'USRTAB',
                    ;        THE USER EXTENDABLE COMMAND TABLE.
                    ;        ($FE AND $FF ARE RESERVED.)
                    ;
                    ; 'CDTAB' IS SEGMENTED FOR ': CONTINUATION' IN RUN MODE.
                    ; ENTRIES BEFORE 'CLNCNT' ALLOW ': CONTINUATION';
                    ; OTHER ENTRIES DO NOT.
                    ;

      = 803E        CDTAB = *          ; COMMAND DATA TABLE BASE ADDRESS.

                    ; ': CONTINUATION' IS VALID IN RUN MODE.

803E  F183          :CDT     DW    XTYPE
8040  6684          :CDY     DW    XTYPE2
8042  6E84          :CDN     DW    XTYPE3
8044  7484          :CDR     DW    XREM

                    ; ': CONTINUATION' IS NOT VALID IN RUN MODE.

      = 0008        CLNCNT   =     *-CDTAB

8046  A790          :CDLST   DW    XLIST
8048  6F91          :CDDEL   DW    XDELET
804A  A688          :CDDMP   DW    XDUMP
804C  F284          :CDRUN   DW    XRUN
      = 0000                 IF    DOS
   -                :CDDOS   DW    XDOS
                             ENDIF
804E  FC8F          :CDLOD   DW    XLOAD
8050  2C90          :CDMRG   DW    XMERGE
8052  3890          :CDAPP   DW    XAPPND

8054  C68F          :CDSAV   DW    XSAVE
8056  1391          :CDAUT   DW    XAUTO
8058  FF91          :CDREN   DW    XREN
805A  1F8A          :CDCOL   DW    XCOLRS
805C  A48A          :CDENS   DW    XENVIR
805E  098A          :CDPAL   DW    XPALET
8060  C38E          :CDDIR   DW    XDIR
8062  068F          :CDCOM   DW    XCOMM
8064  D387          :CDCAL   DW    XCALL
8066  8A8F          :CDTRC   DW    XTRACE
8068  868F          :CDCSS   DW    XCASS
806A  9C8F          :CDSNC   DW    XCSYNC
806C  4385          :CDA     DW    XACCPT
806E  408B          :CDC     DW    XCMPUT
8070  E987          :CDU     DW    XUSE
```

```
8072  C2E8          :CDE     DW   XENE
8074  7E8F          :CDJM    DW   XJMPM
8076  0F88          :CDJ     DW   XJMP
8078  178C          CDG      DW   XGRAPH
807A  2C87          :CDMSX   DW   XMATSX
807C  3287          :CDMS    DW   XMWSP
807E  7086          :CDMX    DW   XMATX
8080  6C86          :CDM     DW   XMATCH
8082  C48C          :CDS     DW   XSOUND
8084  548F          :CDW     DW   XWAIT
8086  8987          :CDNEW   DW   XNEW
8088  AD87          :CDNWV   DW   XNEWV
808A  2188          :CDTV    DW   XTV
808C  6AED          :CDIN    DW   XIN
808E  C48D          :CDOUT   DW   XOUT
8090  F780          :CDDON   DW   XDONE
8092  7484          :CDPOS   DW   XPOS
8094  31A4          :CDSTC   DW   XSETF
8096  26A4          :CDSTL   DW   XSETL
8098  4A90          :CDLTR   DW   XLETTR
809A  778F          :CDSPD   DW   XSPEED
809C  3485          :CDCON   DW   XCONT
809E  ED84          :CDSTP   DW   XSTOP
80A0  5786          :CDAX    DW   XACCX
80A2  6486          :CDAK    DW   XACCK
80A4  7F90          :CDSCR   DW   XSCROLL
80A6  088E          :CDSSA   DW   XSSAV
80A8  5E8E          :CDSLO   DW   XSLOD

      = 006C        TABLEN   SET  *-CDTAB
      = 006C        USROFF   EQU  TABLEN            ; USER TOKENS START AT THIS NUMBER.
      = 0092        USRMAX   EQU  TKNCNT-USROFF     ; USER TABLE OFFSET MAY NOT EXCEED USRMAX.
```

```
                          ; SUBCOMMAND DATA TABLE
                          ;
                          ; CORRESPONDING DATA TABLE
                          ;

        = 80AA            SBDTAB  = *                        ; SUBCOMMAND DATA TABLE BASE ADDRESS.

 80AA  329C               :CDPLS  DW    DADDI
 80AC  429C               :CDSUB  DW    DSUBI
 80AE  879C               :CDDIV  DW    DDIVI
 80B0  549C               :CDMUL  DW    DMULI
 80B2  259D               :CDNE   DW    DNETI
 80B4  3C9D               :CDGE   DW    DGETI
 80B6  439D               :CDLE   DW    DLETI
 80B8  1E9D               :CDEQ   DW    DEQTI
 80BA  359D               :CDLT   DW    DLTTI
 80BC  2C9D               :CDGT   DW    DGTTI
 80BE  E59C               :CDMOD  DW    DMODI
 80C0  689D               :CDAND  DW    DANDI
 80C2  779D               :CDOR   DW    DORI
 80C4  869D               :CDXOR  DW    DXORI
        = 0000                    IF    LOGGRP
   -                      :CDLAN  DW    DLANDI
   -                      :CDLOR  DW    DLORI
                                  ENDIF


 80C6  F19C               :CDUMI  DW    DNEGI
 80C8  959D               :CDNOT  DW    DNOTI
 80CA  579D               :CDLNO  DW    DLNOTI


 80CC  B5A1               :CDDRT  DW    GDRWTO
 80CE  0FA2               :CDDRW  DW    GDRW
 80D0  E9A1               :CDTNT  DW    GTRNTO
 80D2  57A2               :CDTRN  DW    GTRN
 80D4  69A1               :CDGOT  DW    GGOTO
 80D6  13A2               :CDGO   DW    GGO
 80D8  B1A1               :CDFIT  DW    GFILTO
 80DA  0BA2               :CDFIL  DW    GFIL
 80DC  75A2               :CDPEN  DW    GPEN
 80DE  E2A2               :CDCHG  DW    GCHNGE
 80E0  ADA3               :CDCLP  DW    GCLRPN
 80E2  A0A3               :CDCLR  DW    GCLEAR
 80E4  91A3               :CDEXI  DW    GEXIT
 80E6  79A1               :CDFUL  DW    GFULL
 80E8  98A1               :CDSPT  DW    GSPLIT
 80EA  53A3               :CDWAL  DW    GWALL
 80EC  DBA3               :CDEDG  DW    GEDGE
 80EE  B7A3               :CDHOM  DW    GHOME
 80F0  CDA3               :CDNRT  DW    GNORTH
 80F2  F6A3               :CDTRT  DW    GTURTL
 80F4  1AA3               :CDSHD  DW    GSHADE
 80F6  3CA1               :CDMDE  DW    GMODE
 80F8  C6A2               :CDBCK  DW    GBACK
 80FA  F6A1               :CDBK   DW    GBK
```

```
80FC  4CA2        :CDLTU  DW   GLT
80FE  E98C        :CDRPT  DW   GREPT
8100  80A2        :CDPE   DW   GPE
8102  A6A2        :CDPU   DW   GPU
8104  B4A2        :CDPD   DW   GPD


8106  5DB3        :CDRBT  DW   RONOFF
8108  89B3        :CDEYS  DW   REYES
810A  9AB3        :CDRPN  DW   RPEN
810C  B3B3        :CDHRN  DW   RHORN

     = 0064       TABLEN  SET    *-SBDTAB          ; THIS MUST NOT EXCEED 0100 HEX.

                          ASSERT TABLEN<$100
```

```
810E                    PROC
                ;
                ; COND -- CONDITIONAL EXECUTION PROCESSOR
                ;
                ; CALLING SEQUENCE:
                ;
                ;       'INLN' POINTS TO STATEMENT TO BE PROCESSED
                ;       Y = INDEX TO START OF CONDITION.
                ;       'MATCHF' = 0 (FALSE) OR $FF (TRUE) , RESULT OF PRIOR 'M' COMMAND.
                ;
                ;       JSR     COND
                ;
                ;       Y = INDEX TO ':' IN STATEMENT + 1.
                ;       'EXECF' = 0 IF STATEMENT IS NOT TO BE EXECUTED.
                ;
                ; NOTE: GOES TO 'PSTOP' ON ERROR.
                ;
                ; NOTE: 'LOOK AHEAD' CODE FOR GRAPHICS SUBCOMMANDS BEGINNING
                ;       WITH 'Y' OR 'N'.
                ;
810E  A9FF      COND    LDA     #$FF            ; PRESET EXECUTE FLAG.
8110  8D0605            STA     EXECF
8113  20139F            JSR     SLB             ; GET FIRST CHAR OF CONDITION FIELD.

                ; VALID CHARACTERS ARE Y,N,(,:

8116  0920              ORA     #LC             ; FORCE LOWER CASE.
8118  C979              CMP     #'Y'+$20        ; CHECK FOR 'Y' OR 'N' FIRST.
811A  D005 ^8121        BNE     :CN010

811C  A5FE              LDA     MATCHF          ; 'Y' -- IF 'MATCHF' IS TRUE, RESULT IS TRUE.
811E  4C3581            JMP     :CN015

8121  C96E      :CN010  CMP     #'N'+$20
8123  D017 ^813C        BNE     :CN030          ; NOT 'Y' OR 'N'.

                ; SPECIAL CASE '(IMPLIED GR:) NORTH'

8125  C8                INY
8126  B180              LDA     (INLN),Y
8128  88                DEY                     ; POINT INDEX TO 'N'.
8129  0920              ORA     #LC             ; FORCE LOWER CASE.
812B  C96F              CMP     #'O'+$20        ; LOWER CASE 'O'?
812D  F02D ^815C        BEQ     :CN070          ; YES -- TRY 'NORTH'.

812F  A5FE              LDA     MATCHF          ; 'N' -- IF 'MATCHF' IS FALSE, RESULT IS TRUE.
8131  F005 ^8138        BEQ     :CN017

8133  A900              LDA     #0

8135  8D0605    :CN015  STA     EXECF

8138  C8        :CN017  INY
8139  20139F            JSR     SLB             ; GET NEXT NON-BLANK CHARACTER.

813C  B180      :CN030  LDA     (INLN),Y
813E  C928              CMP     #'('            ; SEE IF ARITHMETIC EXPRESSION.
8140  D015 ^8157            BNE     :CN050          ; NO -- ALL DONE.

8142  200FA0                JSR     EXPP            ; EVALUATE EXPRESSION IN PARENS.

8145  A594                  LDA     EXPSTK+1        ; SEE IF RESULT > ZERO.
8147  300C ^814F            BMI     :CN032          ; NO -- NEGATIVE.

8149  D009 ^8154            BNE     :CN040          ; YES -- POSITIVE & NON-ZERO.

814B  A593                  LDA     EXPSTK          ; NOT SURE -- TEST LSB.
814D  D005 ^8154            BNE     :CN040          ; POSITIVE & NON-ZERO.

814F  A900         :CN032   LDA     #0              ; NO -- CONDITION FALSE.
8151  8D0605                STA     EXECF

8154  20139F       :CN040   JSR     SLB             ; GET NEXT NON-BLANK CHARACTER.

8157  C93A         :CN050   CMP     #':'             ; COLON?
8159  D002 ^815D            BNE     :CN080          ; NO.

815B  C8                    INY                     ; SKIP OVER ':'.

815C  60           :CN070   RTS

                   ; ':-REQUIRED' ATTRIBUTE ONLY AVAILABLE DURING SYNTAX CHECK.

815D  A592         :CN080   LDA     EXEC            ; CHECK ': REQUIRED'?
815F  D0FB ^815C            BNE     :CN070          ; NO.
8161  A910                  LDA     #CTCLN          ; ':' REQUIRED FOR THIS COMMAND?
8163  2C1005                BIT     CTABAT
8166  F0F4 ^815C            BEQ     :CN070          ; NO.

8168  88                    DEY
8169  A902                  LDA     #CNDEPR         ; YES -- ERROR.
816B  4C3A7A                JMP     PSTOP
```

```
816E                          PROC
                     ;
                     ; ATOM -- FIND, IDENTIFY & EVALUATE THE NEXT ATOM IN THE STATEMENT LINE.
                     ;
                     ; CALLING SEQUENCE:
                     ;
                     ;        'INLN' POINTS TO THE STATEMENT LINE.
                     ;        Y = INDEX TC END OF PRIOR ATOM + 1.
                     ;
                     ;        JSR     ATOM
                     ;        BNE     SYNTAX ERROR
                     ;
                     ;        A = ATOM IDENTIFIER CODE
                     ;        Y = INDEX TC END OF ATOM + 1 (OR BEGINNING OF ATOM FOR TEXT TYPE).
                     ;        'NUMBER' = VALUE OF NUMERIC CONSTANT OR NUMERIC VARIABLE IF 'EXEC'.
                     ;        'POINT' = ADDRESS OF NUMERIC VARIABLE OR OPERATOR ROUTINE IF 'EXEC'.
                     ;        'NP' POINTS TC STRING VARIABLE NAME.
                     ;        'DP' POINTS TC STRING VARIABLE VALUE (IF DEFINED).
                     ;
816E  20139F         ATOM    JSR     SLB              ; SKIP LEADING BLANKS, IF PRESENT.

                     ; *** INTERNAL RE-ENTRY POINT ***

8171  20F99E         ATOM2   JSR     CHKTRM           ; NULL ATOM (STATEMENT TERMINATOR)?
8174  F036 ^81AE             BEQ     :AT100           ; YES.

8176  C923                   CMP     #'#'              ; NUMERIC VARIABLE?
8178  F039 ^81B3             BEQ     :AT200           ; YES.

817A  C940                   CMP     #'@'              ; POINTER?
817C  D003 ^8181             BNE     :AT002           ; NO.

817E  4C1382                 JMP     :AT250           ; YES.

8181  C924           :AT002  CMP     #'$'              ; STRING VARIABLE?
8183  D003 ^8188             BNE     :AT003           ; NO.

8185  4C5182                 JMP     :AT300           ; YES.

8188  C925           :AT003  CMP     #'%'              ; JOYSTICK/PADDLE/LIGHTPEN?
818A  D003 ^818F             BNE     :AT005           ; NO.

818C  4CC782                 JMP     :AT700           ; YES.

818F  20839E         :AT005  JSR     CNUMBR           ; NUMERIC LITERAL?
8192  B003 ^8197             BCS     :AT010           ; NO

8194  4C9F82                 JMP     :AT400           ; YES.

8197  A200           :AT010  LDX     #OPTABX          ; SPECIAL OPERATOR?
8199  20AE7C                 JSR     SBCMAT
819C  D003 ^81A1             BNE     :AT020           ; NO.

819E  4CAC82                 JMP     :AT600           ; YES.

81A1  B180           :AT020  LDA     (INLN),Y         ; RESTORE CHAR.
81A3  20919E                 JSR     CLETTR           ; CONTEXT DEPENDENT TEXT?
```

```
81A6  B003 ^81AB          BCS     :AT099          ; NO.

81A8  4CA882              JMP     :AT500          ; YES.

81AB  A902        :AT099  LDA     #ATMERR         ; NONE OF THE ABOVE -- ERROR.
81AD  60                  RTS                     ; RETURN WITH CC SET.


                  ; NULL ATOM -- <EOL>

81AE  A901        :AT100  LDA     #NULL
81B0  4CA983              JMP     ATMRET


                  ; NUMERIC VARIABLE -- #<ANY NUMBER OF ALPHANUM>.

81B3  C8          :AT200  INY
81B4  B180                LDA     (INLN),Y        ; CHECK CHARACTER AFTER '#'.
81B6  20869E              JSR     CKEOA
81B9  F0F0 ^81AB          BEQ     :AT099

81BB  A580                LDA     INLN            ; SET NAME POINTER TO NAME.
81BD  85BE                STA     NP
81BF  A581                LDA     INLN+1
81C1  85BF                STA     NP+1
81C3  84C0                STY     NP+2

81C5  20CB9E              JSR     SCEOA           ; SCAN TO END OF ATOM.

                  ; SKIP NUMBERIC VARIABLE LOOKUP IF NOT (EXEC).

81C8  A592                LDA     EXEC
81CA  F043 ^820F          BEQ     :AT220

81CC  84C1                STY     NP+3            ; SAVE LINE INDEX


                  ; ENTRY TO 'FIND' A "JUST - DEFINED" VARIABLE.

81CF  20AD9E      :AT205  JSR     SETSVL          ; SET LIST POINTER TO VARIABLES.
8101  A940                LDA     #ATRNUM         ; 'NUMERIC' ATTRIBUTE.
8103  8D6605              STA     ATRTYP
8106  20CE98              JSR     SFIND           ; FIND VARIABLE IF DEFINED.
8109  F01E ^81F6          BEQ     :AT210          ; DEFINED.

810B  A900                LDA     #0              ; DATA = 00.
810D  8586                STA     NUMBER
810F  8587                STA     NUMBER+1
81E1  85C4                STA     DP+2
                  ; *5*   LDA     # HIGH NUMBER
81E3  85C3                STA     DP+1
81E5  A986                LDA     # LOW NUMBER
81E7  85C2                STA     DP
81E9  A902                LDA     #2
81EB  85C5                STA     DP+3
81ED  200599              JSR     SINSRT          ; INSERT NUMERIC (VALUE = 0).
```

```
81F0  D0B9 ^81AB          BNE     :AT099          ; ERROR -- NO ROOM.

81F2  A4C1                LDY     NP+3            ; SEARCH AGAIN -- FIND IT.
81F4  D0D8 ^81CE          BNE     :AT205          ; (BRA).

81F6  A236       :AT210   LDX     #POINT-DTAB     ; ADDRESS OF VALUE ...
81F8  A042                LDY     #DP-DTAB
81FA  20459A              JSR     DMOVI
81FD  A5C4                LDA     DP+2
81FF  20049D              JSR     DADDS           ; ... IN 'POINT'.

8202  A000                LDY     #0              ; VALUE IN 'NUMBER'.
8204  B1B6                LDA     (POINT),Y
8206  85E8                STA     NUMBER
8208  C8                  INY
8209  B1B6                LDA     (POINT),Y
820B  85E9                STA     NUMBER+1

820D  A4C1                LDY     NP+3            ; RESTORE LINE INDEX.

820F  A904       :AT220   LDA     #NVAR
8211  D072 ^8285          BNE     :AT340          ; (BRA) TO 'ATMRET'.


                 ; POINTER (INDIRECT REFERENCE) -- @[B]<NUMERIC QUANTITY>

8213  C8         :AT250   INY                     ; EXAMINE CHARACTER AFTER '@'.
8214  B180                LDA     (INLN),Y
8216  0920                ORA     #LC             ; FORCE LOWER CASE.
8218  C962                CMP     #'B'+$20        ; POINTER TO BYTE?
821A  08                  PHP                     ; SAVE ANSWER.
821B  D001 ^821E          BNE     :AT255          ; NO -- POINTER TO WORD.

821D  C8                  INY                     ; YES -- SKIP OVER 'B'.

821E  B180       :AT255   LDA     (INLN),Y        ; GET CHARACTER FOR RECURSIVE CALL.

8220  207181              JSR     ATOM2           ; SEE WHAT FOLLOWS *** RECURSIVE CALL ***.
8223  D028 ^824D          BNE     :AT290          ; ERROR.

8225  2906                AND     #NVAR+NUM       ; MUST BE NUMERIC.
8227  F024 ^824D          BEQ     :AT290          ; ERROR.

8229  A5E8                LDA     NUMBER          ; RESULT IS ADDRESS OF DATA.
822B  85B6                STA     POINT
822D  A5E9                LDA     NUMBER+1
822F  85E7                STA     POINT+1

8231  84A1                STY     TEMP            ; SAVE LINE INDEX.
8233  A000                LDY     #0              ; GET DATA VALUE NOW.
8235  B1B6                LDA     (POINT),Y
8237  85E8                STA     NUMBER
8239  28                  PLP                     ; POINTER TO BYTE?
823A  D006 ^8242          BNE     :AT260          ; NO -- POINTER TO WORD.

823C  84E9                STY     NUMBER+1        ; YES -- MSB = 0.
```

```
823E  A980                  LDA     #BPTR           ; TYPE = POINTER TO BYTE.
8240  D007 ^8249            BNE     :AT270          ; (BRA).

8242  C8          :AT260    INY
8243  B186                  LDA     (POINT),Y       ; GET MSB OF DATA WORD.
8245  8589                  STA     NUMBER+1
8247  A904                  LDA     #NVAR           ; TYPE = POINTER TO WORD.

8249  A4A1        :AT270    LDY     TEMP            ; RESTORE LINE INDEX.
824B  D038 ^8285            BNE     :AT340          ; (BRA) SKIP TO NORMAL RETURN.

824D  28          :AT290    PLP                     ; CLEANUP STACK BEFORE RETURN.
824E  4C4B81      :AT299    JMP     :AT099          ; ERROR RETURN (EXTENDED BRANCH).



                  ; STRING VARIABLE -- $<ANY NUMBER OF ALPHANUM>

8251  C8          :AT300    INY                     ; EXAMINE CHARACTER AFTER '$'.
8252  B180                  LDA     (INLN),Y
8254  C924                  CMP     #'$'            ; STRING INDIRECTION?
8256  F030 ^8288            BEQ     :AT350          ; YES.

8258  20B69E                JSR     CKEDA           ; NO -- STRING NAME ERROR?
825B  F0F1 ^824E            BEQ     :AT299          ; YES.

825D  A580                  LDA     INLN            ; NO -- SET NAME POINTER TO NAME.
825F  858E                  STA     NP
8261  A581                  LDA     INLN+1
8263  858F                  STA     NP+1
8265  84C0                  STY     NP+2

8267  20C69E                JSR     SCECA           ; SCAN TO END OF ATOM.
826A  84C1                  STY     NP+3            ; SAVE END INDEX.

826C  98          :AT320    TYA                     ; SAVE LINE INDEX.
826D  48                    PHA
826E  20A09E                JSR     SETSVL          ; SET LIST POINTER TO STRING VARIABLES.
8271  A9F0                  LDA     #ATRSTR         ; 'STRING' ATTRIBUTE.
8273  8D6605                STA     ATRTYP
8276  20CE98                JSR     SFIND           ; FIND VARIABLE IF DEFINED.
8279  D006 ^8281            BNE     :AT330          ; UNDEFINED.

827B  68                    PLA                     ; RESTORE LINE INDEX.
827C  A8                    TAY
827D  A908                  LDA     #SVAR           ; DEFINED STRING VARIABLE.
827F  D004 ^8285            BNE     :AT340          ; (BRA) TO 'ATMRET'.

8281  68          :AT330    PLA                     ; RESTORE LINE INDEX.
8282  A8                    TAY
8283  A910                  LDA     #USVAR          ; UNDEFINED STRING VARIABLE.

8285  4C0983      :AT340    JMP     ATMRET          ; *** SKIP BRANCH POINT ***

8288  205182      :AT350    JSR     :AT300          ; INDIRECTION -- GET NAME *** RECURSIVE CALL ***.
828B  D011 ^829E            BNE     :AT360          ; ERROR.
```

```
828D  C910                 CMP     #USVAR          ; UNDEFINED STRING?
828F  F0F4 ^8285           BEQ     :AT340          ; YES -- ALL DONE.

8291  84A1                 STY     TEMP            ; DEFINED -- USE DATA AS NAME FOR TARGET.
8293  A23E                 LDX     #NP-DTAB
8295  A042                 LDY     #DP-DTAB
8297  203B9A               JSR     PMOVE
829A  A4A1                 LDY     TEMP
829C  D0CE ^826C           BNE     :AT320          ; (BRA) NOW GET STRING.

829E  60          :AT360   RTS



                  ; NUMERIC LITERAL -- <DIGIT><ANY NUMBER OF DIGITS>

829F  A200        :AT400   LDX     #INLN-DTAB      ; POINT TO POINTER.
82A1  20BB9D               JSR     ASCDEC          ; CONVERT TO BINARY, RESULT TO 'NUMBER'.

82A4  A902                 LDA     #NUM
82A6  D0DD ^8285           BNE     :AT340          ; (BRA) TO 'ATMRET'.



                  ; TEXT -- <LETTER><ANY NUMBER OF CHARACTERS>

82A8  A920        :AT500   LDA     #TEXT
82AA  D0D9 ^8285           BNE     :AT340          ; (BRA) TO 'ATMRET'.

                  ; OPERATOR -- <OPERATOR>


82AC  BDAA80      :AT600   LDA     SBDTAB,X        ; GET OPERATE ROUTINE ADDRESS.
82AF  8586                 STA     POINT
82B1  BDAB80               LDA     SBDTAB+1,X
82B4  8587                 STA     POINT+1
82B6  A940                 LDA     #OPR
82B8  D0CB ^8285           BNE     :AT340          ; (BRA) TO 'ATMRET'.


                  ; EVALUATE EXPRESSION -- %(<NEXP>)

82BA  200FA0      :AT620   JSR     EXPP            ; EVALUATE NEXP IN PARENS.
82BD  A593                 LDA     EXPSTK          ; PASS BACK RESULT.
82BF  A694                 LDX     EXPSTK+1
82C1  4C6283               JMP     :AT781

                  ; CONTROLLERS -- %<P!J!T><NUMBER>  OR  %<X!Y!Z!A!H!V!L!M!F>  OR  %<S!SR!ST>


82C4  4CAB81      :AT720   JMP     :AT099          ; ERROR.

82C7  C8          :AT700   INY                     ; SKIP OVER '%'.
82C8  B180                 LDA     (INLN),Y        ; GET NEXT CHARACTER.
82CA  C928                 CMP     #'(             ; EVAL?
82CC  F0EC ^82BA           BEQ     :AT620          ; YES.
```

```
82CE  0920                    ORA    #LC             ; FORCE LOWER CASE.
82D0  C970                    CMP    #'P'+$20        ; PADDLE CONTROLLER?
82D2  F035 ^8309              BEQ    :AT730          ; YES.

82D4  C96E                    CMP    #'N'+$20        ; PEN NUMBER?
82D6  D003 ^82DB              BNE    :AT703          ; NO.
82D8  4CDA83                  JMP    :AT960          ; YES.

82DB  C96B           :AT703   CMP    #'K'+$20        ; KEY PRESS READ?
82DD  F03B ^831A              BEQ    :AT735          ; YES.

82DF  C966                    CMP    #'F'+$20        ; FREE MEMORY?
82E1  F045 ^8328              BEQ    :AT740          ; YES.

82E3  C96A                    CMP    #'J'+$20        ; JOYSTICK?
82E5  F05A ^8341              BEQ    :AT760          ; YES.

82E7  C974                    CMP    #'T'+$20        ; TRIGGER?
82E9  F066 ^8351              BEQ    :AT770          ; YES.

82EB  C973                    CMP    #'S'+$20        ; TURTLE SENSORS?
82ED  D003 ^82F2              BNE    :AT705          ; NO.
82EF  4CAE83                  JMP    :AT950          ; YES.

82F2  C978           :AT705   CMP    #'X'+$20        ; GRAPHICS X?
82F4  F074 ^836A              BEQ    :AT782          ; YES.

82F6  C979                    CMP    #'Y'+$20        ; GRAPHICS Y?
82F8  F074 ^836E              BEQ    :AT784          ; YES.

82FA  C97A                    CMP    #'Z'+$20        ; GRAPHICS PIXEL VALUE.
82FC  F07B ^8379              BEQ    :AT788          ; YES.

82FE  C961                    CMP    #'A'+$20        ; GRAPHICS THETA ANGLE?
8300  F070 ^8372              BEQ    :AT786          ; YES.

      = 0000                  IF     LITPEN
  -                           CMP    #'H'+$20        ; LIGHTPEN HORIZONTAL?
  -                           BEQ    :AT790          ; YES.

  -                           CMP    #'V'+$20        ; LIGHTPEN VERTICAL?
  -                           BEQ    :AT795          ; YES.

  -                           CMP    #'L'+$20        ; LIGHTPEN TRIGGER?
  -                           BEQ    :AT796          ; YES.
                              ENDIF

8302  C96D                    CMP    #'M'+$20        ; MATCH RESULT?
8304  D0BE ^82C4              BNE    :AT720          ; NO.

8306  4C8083                  JMP    :AT798          ; YES.

                     ; READ PADDLE CONTROLLER

8309  20B583         :AT730   JSR    :AT800          ; GET VALUE THAT FOLLOWS 'P'.
830C  D036 ^8344              BNE    :AT761          ; ERROR.
```

```
830E  2907                AND    #$07           ; PADDLE # MODULO 8.
8310  AA                  TAX
8311  38                  SEC                   ; (CLEAR BORROW).
8312  A9E4                LDA    #228           ; RESULT = 228 - VALUE READ.
8314  FD7002              SBC    PADDL0,X
8317  4C6083              JMP    :AT780

831A  C8          :AT735  INY                   ; SKIP OVER 'K'.
831B  ADFC02              LDA    CH             ; KEYCODE READY?
831E  38                  SEC
831F  E9FF                SBC    #$FF
8321  F002 ^8325          BEQ    :AT737         ; NO.

8323  A901                LDA    #1             ; YES.

8325  4C6083      :AT737  JMP    :AT780

                  ; CALCULATE FREE MEMORY

8328  C8          :AT740  INY                   ; SKIP OVER 'F'.
8329  38                  SEC
832A  A582                LDA    S2L            ; 'NUMBER' = 'S2L' - 'S1H' + 1.
832C  E580                SBC    S1H
832E  85B8                STA    NUMBER
8330  A583                LDA    S2L+1
8332  E581                SBC    S1H+1
8334  85B9                STA    NUMBER+1
8336  E6B8                INC    NUMBER
8338  D002 ^833C          BNE    :AT745
833A  E6B9                INC    NUMBER+1
833C  A902        :AT745  LDA    #NUM           ; TYPE = NUMBER.
833E  4CA983              JMP    ATMRET

                  ; READ JOYSTICK

8341  208583      :AT760  JSR    :AT800         ; GET VALUE THAT FOLLOWS 'J'.
8344  D04D ^8393  :AT761  BNE    :AT890         ; ERROR *** SKIP BRANCH POINT ***

8346  2903                AND    #$03           ; JOYSTICK # MODULO 4.
8348  AA                  TAX
8349  BD7802              LDA    STICK0,X       ; GET JOYSTICK DATA FROM DATA BASE.
834C  490F                EOR    #$0F           ; INVERT DATA READ.
834E  4C6083              JMP    :AT780

                  ; READ TRIGGER

8351  208583      :AT770  JSR    :AT800         ; GET VALUE THAT FOLLOWS 'T'.
8354  D03D ^8393          BNE    :AT890         ; ERROR.

8356  290F                AND    #$0F           ; TRIGGER # MODULO 16.
8358  AA                  TAX
8359  BD7C02              LDA    PTRIG0,X       ; RESULT = SINGLE BIT.
835C  49FF                EOR    #$FF
835E  2901                AND    #$01

                  ; *** ENTRY FOR TURTLE SENSORS ***.
```

```
8360  A200         :AT780  LDX     #0              ; M.S.B. = 0.

8362  85E8         :AT781  STA     NUMBER          ; STORE RESULT.
8364  86E9                 STX     NUMBER+1
8366  A902                 LDA     #NUM            ; NUMERIC RESULT.
8368  D03F ^83A9           BNE     ATMRET          ; (BRA).

                   ; GRAPHICS PARAMETERS

836A  A26C         :AT782  LDX     #GX-DTAB        ; GRAPHICS X COORDINATE.
836C  B029 ^8397           BCS     :AT900          ; (BRA).

836E  A26F         :AT784  LDX     #GY-DTAB        ; GRAPHICS Y COORDINATE.
8370  B025 ^8397           BCS     :AT900          ; (BRA).

8372  A5F2         :AT786  LDA     THETA           ; GRAPHICS THETA ANGLE.
8374  A6F3                 LDX     THETA+1
8376  C8                   INY
8377  D0E9 ^8362           BNE     :AT781          ; (BRA).

8379  C8           :AT788  INY
837A  205EAC               JSR     GREAD           ; READ GRAPHICS DATA.
837D  4C6083               JMP     :AT780

        = 0000             IF      LITPEN
                   ; READ LIGHTPEN

  -                :AT790  LDA     LPENH           ; LIGHTPEN HORIZONTAL VALUE.
  -                        BCS     :AT797          ; (BRA).

  -                :AT795  LDA     LPENV           ; LIGHTPEN VERTICAL VALUE.
  -                        BCS     :AT797          ; (BRA).

  -                :AT796  LDA     STICK0          ; GET LIGHTPEN TRIGGER.
  -                        EOR     #$01            ; INVERT BIT OF INTEREST.
  -                        AND     #$01

  -                :AT797  LDX     EXEC            ; EXECUTE MODE?
  -                        BEQ     :AT798          ; NO.

  -                        LDX     #$0A            ; BACKGROUND = LIGHT GRAY.
  -                        STX     COLCR0+4

  -                :AT798  INY
  -                        BNE     :AT780          ; (BRA).
                           ENDIF

                   ; READ MATCH FLAG

8380  A5FE         :AT798  LDA     MATCHF          ; MATCH RESULT FLAG.
8382  C8                   INY
8383  D0DB ^8360           BNE     :AT780          ; (BRA).

                   ; SUBROUTINE TO PROCESS NUMBER FOLLOWING %P, %J & %T.

8385  C8           :AT800  INY                     ; SKIP OVER 'P' OR 'J' OR 'T'.
8386  206E81               JSR     ATON            ; *** RECURSIVE CALL ***.
```

```
8389  D009 ^8394            BNE     :AT895            ; ERROR.

838B  2906                  AND     #NVAR+NUM         ; NUMERIC RESULT?
838D  F005 ^8394            BEQ     :AT895            ; NO -- ERROR.

838F  A588                  LDA     NUMBER            ; YES.
8391  C588                  CMP     NUMBER            ; SET CC FOR NORMAL EXIT.

8393  60           :AT890   RTS                       ; RETURN WITH CC SET.

8394  A902         :AT895   LDA     #ATMERR           ; INVALID # AFTER LETTER.
8396  60                    RTS                       ; RETURN WITH CC SET.

                   ; SUBROUTINE TO ROUND & STORE THE GRAPHICS COORDINATES

8397  C8           :AT900   INY
8398  B582                  LDA     DTAB+2,X          ; GET FRACTIONAL PORTION.
839A  2A                    ROL     A                 ; MSB OF FRACTION TO CARRY BIT.
839B  B580                  LDA     DTAB+0,X          ; ROUND LSB.
839D  6900                  ADC     #0
839F  8588                  STA     NUMBER
83A1  B581                  LDA     DTAB+1,X          ; CARRY TO MSB.
83A3  6900                  ADC     #0
83A5  8589                  STA     NUMBER+1
83A7  A902                  LDA     #NUM              ; NUMERIC RESULT.


83A9  85A1         ATMRET   STA     TEMP              ; SET CC FOR EXIT.
83AB  C5A1                  CMP     TEMP
83AD  60                    RTS

                   ; TURTLE SENSORS
                   ;
                   ; %S  = ROBOT IF ON, ELSE VISIBLE TURTLE.
                   ; %SR = ROBOT.
                   ; %ST = VISIBLE TURTLE.

83AE  C8           :AT950   INY                       ; SKIP OVER 'S'.
83AF  B180                  LDA     (INLN),Y          ; GET NEXT CHARACTER.
83B1  C8                    INY                       ; SKIP OVER NEXT CHARACTER.
83B2  0920                  ORA     #LC               ; FORCE LOWER CASE.
83B4  C972                  CMP     #'R'+$20          ; %SR?
83B6  F00A ^83C2            BEQ     :AT952            ; YES.
83B8  C974                  CMP     #'T'+$20          ; %ST?
83BA  F011 ^83CD            BEQ     :AT954            ; YES.

83BC  88                    DEY                       ; %S.
83BD  ADC505                LDA     RBTON             ; ROBOT OR VISIBLE?
83C0  F00B ^83CD            BEQ     :AT954            ; VISIBLE.

83C2  ADC505       :AT952   LDA     RBTON             ; SENSORS = 0 IF ROBOT OFF.
83C5  F099 ^8360            BEQ     :AT780            ; OFF.
83C7  200DB4                JSR     RRDSNS            ; ROBOT.
83CA  4C6083                JMP     :AT780

83CD  A592         :AT954   LDA     EXEC              ; EXECUTE MODE?
83CF  F006 ^83D7            BEQ     :AT956            ; NO.
```

```
83D1  2098AC                JSR     VTSENS
83D4  AD5005                LDA     TRTSNS           ; VISIBLE.

83D7  4C6083       :AT956   JMP     :AT780

                   ; %N = TURTLE PEN NUMBER

83DA  C8           :AT960   INY                      ; SKIP OVER 'N'.
83DB  AD1305                LDA     PEN              ; GET PEN #.
83DE  4C6083                JMP     :AT780
```

```
83E1                        PROC
                    ;
                    ; XTYPE -- TYPE COMMAND PROCESSOR
                    ;
83E1  20A7A0        XTYPE   JSR     TEXP            ; PROCESS TEXT EXPRESSION.

83E4  A592                  LDA     EXEC            ; EXECUTE MODE?
83E6  F07D ^8465            BEQ     :XT090          ; NO.

83E8  20B896                JSR     TSTMOD          ; CHECK SCREEN MODE.
83EB  C908                  CMP     #GRFS           ; FULL SCREEN GRAPHICS.
83ED  D003 ^83F2            BNE     :XT005          ; NO.

83EF  A983                  LDA     #NPCERR         ; YES -- ERROR.
83F1  60                    RTS

83F2  A68F          :XT005  LDX     TELN+3          ; CHECK FOR NULL TEXT.
83F4  F00C ^8402            BEQ     :XT010          ; NULL.

83F6  BDFF8B                LDA     TEXBUF-1,X      ; NON-NULL -- CHECK FINAL CHARACTER.
83F9  C95C                  CMP     #BSLASH         ; IS IT EOL SUPPRESS?
83FB  D005 ^8402            BNE     :XT010          ; NO.

83FD  C68F                  DEC     TELN+3          ; YES -- SUPPRESS '\' ALSO.
83FF  4C0984                JMP     :XT020

8402  A99B          :XT010  LDA     #EOL            ; INSERT EOL.
8404  9D008C                STA     TEXBUF,X
8407  E68F                  INC     TELN+3

                    ; TYPE WITH WORD SPLIT AVOIDANCE.

8409  84AC          :XT020  STY     XTEMP+1         ; SAVE STATEMENT INDEX.
840B  A48E                  LDY     TELN+2          ; STARTING INDEX.
840D  C48F                  CPY     TELN+3
840F  F050 ^8461            BEQ     :XT080          ; NULL OUTPUT -- ALL DONE.

8411  84AB          :XT022  STY     XTEMP           ; SAVE INDEX.
8413  A655                  LDX     COLCRS          ; GET CURRENT CURSOR POSITION.
8415  AD1405                LDA     GRFLAG          ; DIFFERENT CURSOR IF SPLIT SCREEN.
8418  F003 ^841D            BEQ     :XT025

841A  AE9102                LDX     TXTCOL          ; SPLIT SCREEN -- USE OTHER CURSOR.

841D  86AD          :XT025  STX     XTEMP+2         ; SAVE STARTING COLUMN #.
841F  CA                    DEX                     ; PRE-CONDITION THE INDEX.

8420  B18C          :XT030  LDA     (TELN),Y        ; FIND LENGTH OF NEXT WORD.
8422  E8                    INX
8423  C8                    INY
8424  C48F                  CPY     TELN+3
8426  F004 ^842C            BEQ     :XT035          ; END OF TEXT.

8428  C920                  CMP     #' '             ; SPACE?
842A  D0F4 ^8420            BNE     :XT030          ; NO -- KEEP SCANNING.
```

```
                    ATARI CAMAC Assembler Ver  1.0A  Page  59
PILOT -- H.G. STEWART                                        D1:PILOT.

842C  A4AB         :XT035  LDY     XTEMP           ; END OF WORD -- CHECK FOR WORD SPLIT.
842E  ECFE05               CPX     RGCOL           ; DOES IT WRAP SCREEN?
8431  F00C ^843F           BEQ     :XT040          ; NO -- OUTPUT IT.
8433  900A ^843F           BCC     :XT040          ; NO -- OUTPUT IT.

8435  A5AD                 LDA     XTEMP+2         ; YES -- IS THIS THE 1ST WORD OF LINE?
8437  CDF505               CMP     LFCOL
843A  F003 ^843F           BEQ     :XT040          ; YES -- FORGET ABOUT NEW LINE.

843C  20969F               JSR     NEWLIN          ; NO -- START A NEW LINE.

843F  B18C         :XT040  LDA     (TELN),Y        ; OUTPUT THE WORD JUST SCANNED.
8441  C920                 CMP     #' '             ; SPACE?
8443  D007 ^844C           BNE     :XT050          ; NO.

8445  ECFE05               CPX     RGCOL           ; YES -- IS IT THE LAST POSITION?
8448  D002 ^844C           BNE     :XT050          ; NO.

844A  A99B                 LDA     #EOL            ; YES -- CHANGE TO EOL.

844C  208294       :XT050  JSR     CHOT            ; OUTPUT CHAR.
844F  2048AA               JSR     SPDDEL          ; DELAY IF SPECIFIED.

8452  C8                   INY
8453  C48F                 CPY     TELN+3          ; END OF TEXT?
8455  F00A ^8461           BEQ     :XT080          ; YES.

8457  88                   DEY
8458  B18C                 LDA     (TELN),Y        ; SPACE?
845A  C8                   INY
845B  C920                 CMP     #' '             ; SPACE?
845D  D0E0 ^843F           BNE     :XT040          ; NO -- NOT END OF WORD.

845F  F0B0 ^8411           BEQ     :XT022          ; YES -- NOW DO NEXT WORD (BRA).

8461  A4AC         :XT080  LDY     XTEMP+1         ; RESTORE STATEMENT INDEX.
8463  A900                 LDA     #0              ; SET CC FOR EXIT.

8465  60           :XT090  RTS                     ; RETURN WITH CC SET.


                   ; 'Y' COMMAND PROCESSOR

8466  F00F ^8477   XTYPE2  BEQ     :XT500          ; SYNTAX SCAN ONLY.

8468  A5FE                 LDA     MATCHF          ; Y COMMAND (SAME AS 'TY').
846A  D00B ^8477           BNE     :XT500

846C  F006 ^8474           BEQ     :XT400

                   ; 'N' COMMAND PROCESSOR

846E  F007 ^8477   XTYPE3  BEQ     :XT500          ; SYNTAX SCAN ONLY.

8470  A5FE                 LDA     MATCHF          ; N COMMAND (SAME AS 'TN').
8472  F003 ^8477           BEQ     :XT500          ; SKIP BRANCH TO 'XTYPE'.
```

```
846E  F007 ^8477  XTYPE3  BEQ     :XT500          ; SYNTAX SCAN ONLY.

8470  A5FE                LDA     MATCHF          ; N COMMAND (SAME AS 'TN').
8472  F003 ^8477          BEQ     :XT500          ; SKIP BRANCH TO 'XTYPE'.
```

```
8474              XREM                            ; REMARK COMMAND PROCESSOR TOO.

8474  4C169F      :XT400  JMP     SCNEOL          ; SCAN TO END OF LINE & RETURN WITH CC SET.

8477  4CE183      :XT500  JMP     XTYPE
```

```
847A                   PROC
                 ;
                 ; XPOS -- POSITION COMMAND PROCESSOR
                 ;
847A  20049F     XPOS    JSR    EXP            ; COLUMN NUMBER.
847D  A592               LDA    EXEC           ; EXECUTE MODE?
847F  F019 ^849A         BEQ    :XP020         ; NO.

8481  208B96             JSR    TSTMOD         ; CHECK SCREEN MODE.
8484  C908               CMP    #GRFS          ; FULL GRAPHICS?
8486  F034 ^84BC         BEQ    :XP080         ; YES -- IGNORE COMMAND.

8488  A594               LDA    EXPSTK+1       ; RANGE CHECK THE COLUMN #.
848A  D033 ^84BF         BNE    :XP900         ; TOO LARGE.

848C  A593               LDA    EXPSTK+0       ; PAST RIGHT MARGIN?
848E  CD8605             CMP    RGCOL
8491  F002 ^8495         BEQ    :XP010
8493  B02A ^84BF         BCS    :XP900         ; YES -- TOO LARGE.

8495  8555       :XP010  STA    COLCRS         ; O.K. -- STORE IT.
8497  8D9102             STA    TXTCOL         ; SPLIT SCREEN TOO.

849A  20079F     :XP020  JSR    SKPSEP         ; SKIP SEPARATOR.
849D  20049F             JSR    EXP            ; ROW NUMBER.
84A0  A592               LDA    EXEC           ; EXECUTE MODE?
84A2  F01A ^84BE         BEQ    :XP090         ; NO.

84A4  AD3505             LDA    TRACE          ; TRACE EXECUTION?
84A7  0D4505             ORA    SGLSTP
84AA  D010 ^84BC         BNE    :XP080         ; YES -- IGNORE THIS COMMAND.

84AC  A594               LDA    EXPSTK+1       ; RANGE CHECK THE ROW #.
84AE  D00F ^84BF         BNE    :XP900         ; TOO LARGE.

84B0  A593               LDA    EXPSTK+0
84B2  CD8F02             CMP    BOTSCR
84B5  B008 ^84BF         BCS    :XP900         ; TOO LARGE.

84B7  8554               STA    ROWCRS         ; O.K. -- STORE IT.
84B9  8D9002             STA    TXTROW         ; SPLIT SCREEN TOO.

84BC  A900       :XP080  LDA    #0             ; SET CC FOR NORMAL EXIT.

84BE  60         :XP090  RTS                   ; RETURN WITH CC SET.

84BF  A902       :XP900  LDA    #IMPERR        ; COLUMN/ROW OUT OF RANGE.
84C1  60                 RTS                   ; RETURN WITH CC SET.
```

```
84C2                        PROC
                    ;
                    ; XEND -- END STATEMENT PROCESSOR
                    ;
84C2  F016 ^84DA    XEND    BEQ     :XE090          ; SYNTAX SCAN ONLY.

84C4  AE4D05                LDX     USTKP           ; USE STACK POINTER.
84C7  F014 ^84DD            BEQ     :XE095          ; STACK EMPTY.

84C9  86FF                  STX     RUN             ; SET RUN MODE EVEN IF ALREADY SET.
84CB  CA                    DEX                     ; GET NEXT LINE ADDRESS FROM STACK.
84CC  CA                    DEX
84CD  8E4D05                STX     USTKP
84D0  BD6B05                LDA     USESTK,X
84D3  8584                  STA     NXTLN
84D5  BD6C05                LDA     USESTK+1,X
84D8  8585                  STA     NXTLN+1

84DA  A900          :XE090  LDA     #0              ; O.K. -- SET CC FOR EXIT.

84DC  60                    RTS

84DD  208198        :XE095  JSR     CLOSEM          ; CLOSE ALL OPEN FILES.
84E0  84AB                  STY     XTEMP
84E2  209E98                JSR     REMDEV
84E5  A4AB                  LDY     XTEMP
84E7  A981                  LDA     #ENDERR         ; STOP CONDITION.
84E9  8D4305                STA     NOCONT          ; NO CONTINUE AFTER END.
84EC  60                    RTS

84ED                        PROC
                    ;
                    ; XSTOP -- STOP COMMAND PROCESSOR
                    ;
84ED  F002 ^84F1    XSTOP   BEQ     :XS090          ; SYNTAX SCAN ONLY.

84EF  A99A                  LDA     #STPMES         ; GENERATE STOP MESSAGE.

84F1  60            :XS090  RTS
```

```
      = 0000               IF      DOS
   -                       PROC
                   ;
                   ; XDOS -- DOS COMMAND PROCESSOR
                   ;
   -               XDOS    BEQ     :XD090          ; SYNTAX SCAN ONLY.

   -                       STA     COLDST          ; SETUP FOR COLDSTART ON RESET.
   -                       JSR     TXOPEN          ; OPEN TEXT SCREEN.
   -                       JMP     (DSVSAV)        ; YES.

   -               :XD090  RTS
                           ENDIF
```

```
84F2                        PROC
                    ;
                    ; XRUN -- RUN COMMAND PROCESSOR
                    ;
84F2  20139F        XRUN    JSR     SLB             ; 'RUN <EOL>'?
84F5  20F99E                JSR     CHKTRM
84F8  F011 ^850B            BEQ     :XR005          ; YES.

                    ; ASSUME 'RUN <FILE>' - SHARP 'LOAD' CODE.

84FA  201090                JSR     XL0100          ; OPEN DEVICE.
84FD  D034 ^8533            BNE     :XR090          ; ERROR.

84FF  A592                  LDA     EXEC            ; EXECUTE MODE?
8501  F005 ^8508            BEQ     :XR003          ; NO.

8503  85FF                  STA     RUN             ; YES -- SET RUN MODE.
8505  201485                JSR     :XR020          ; INITIALIZE ENVIRONMENT.

8508  4C0190        :XR003  JMP     XL0005          ; NOW LET LOAD DO THE SETUP.

850B  A592          :XR005  LDA     EXEC            ; EXECUTE MODE?
850D  F024 ^8533            BEQ     :XR090          ; NO.

850F  85FF                  STA     RUN             ; YES -- ENTER RUN MODE.

                    ; *** EXTERNAL ENTRY POINT FROM 'MLOOP' ***

8511  20A087        XRN010  JSR     XNEWV           ; CLEAR ALL VARIABLES.

8514  A5AE          :XR020  LDA     S1L             ; SETUP THE NEXT LINE POINTER.
8516  8584                  STA     NXTLN
8518  A5AF                  LDA     S1L+1
851A  8585                  STA     NXTLN+1

851C  20A0A3                JSR     GCLEAR          ; CLEAR SCREEN.
851F  2060AF                JSR     GPINIT          ; INITIALIZE GRAPHICS PARAMETERS.

8522  84AB                  STY     XTEMP
8524  20729F                JSR     NULACC          ; SET ACCEPT BUFFER TO NULL.
8527  A4AB                  LDY     XTEMP

8529  A900                  LDA     #0              ; MAKE MATCH FLAG FALSE ...
852B  8D4D05                STA     USTKP           ; ... USE STACK INDEX ...
852E  85FE                  STA     MATCHF          ; ... & SET CC ALSO.
8530  8D4305                STA     NOCONT          ; CONTINUE O.K.

8533  60            :XR090  RTS

8534                        PROC
                    ;
                    ; XCONT -- CONTINUE COMMAND PROCESSOR
                    ;
8534  F009 ^853F    XCONT   BEQ     :XC090          ; SYNTAX SCAN ONLY.

8536  AE4305                LDX     NOCONT          ; CONTINUE O.K.?
8539  D005 ^8540            BNE     :XC100          ; NO -- INFORM OPERATOR.
```

```
8538  85FF                 STA     RUN           ; YES -- ENTER RUN MODE.
853D  A900                 LDA     #0            ; SET CC FOR NORMAL RETURN.

853F  60          :XC090   RTS

8540  A999        :XC100   LDA     #CNTERR       ; CONTINUE ERROR.
8542  60                   RTS
```

```
8543                     PROC
                 ;
                 ; XACCPT -- ACCEPT COMMAND PROCESSOR
                 ;
8543  A900       XACCPT  LDA    #0              ; STANDARD ACCEPT.
8545  8D4705             STA    AKFLAG
8548  8D4605             STA    AXFLAG

                 ; *** EXTERNAL ENTRY POINT FROM 'XACCX' AND 'XACCK' ***

854B  20009F     :XA001  JSR    CHKEQS          ; '='?
854E  D006 ^8556         BNE    :XA003          ; NO OR NOT YET.

8550  A901               LDA    #NULL           ; SETUP FOR NULL TARGET.
8552  85AB               STA    XTEMP
8554  D051 ^85A7         BNE    :XA022          ; (BRA).

8556  206E81     :XA003  JSR    ATOM            ; CHECK FOR VARIABLE.
8559  D008 ^8563         BNE    :XA009          ; ERROR.

855B  85AB               STA    XTEMP           ; SAVE ATOM TYPE.
855D  299D               AND    #SVAR+USVAR+NVAR+NULL+BPTR
855F  D003 ^8564         BNE    :XA020          ; VALID ATOM TYPE.

8561  A902               LDA    #IMPERR         ; NONE OF THE ABOVE -- ERROR.

8563  60         :XA009  RTS                    ; RETURN WITH CC SET.

8564  208B8B     :XA020  JSR    SAVIT           ; YES -- SAVE NAME IF STRING TARGET.

8567  20009F     :XA20D  JSR    CHKEQS          ; CHECK FOR ASSIGNMENT OPTION.
856A  F03B ^85A7         BEQ    :XA022          ; YES.

856C  A592               LDA    EXEC            ; EXECUTE MODE?
856E  F0F3 ^8563         BEQ    :XA009          ; NO.

8570  84AC               STY    XTEMP+1         ; SAVE STATEMENT INDEX.
8572  AD4705             LDA    AKFLAG          ; ACCEPT KEY?
8575  D017 ^858E         BNE    :XA021          ; YES.

8577  208B96             JSR    TSTMOD          ; CHECK SCREEN MODE.
857A  2905               AND    #TXSL+GRSS      ; TEXT INPUT O.K.?
857C  D003 ^8581         BNE    :XA20G          ; YES.

857E  A983               LDA    #NRCERR         ; NO -- ERROR.
8580  60                 RTS

8581  A20C       :XA20G  LDX    #TELN-DTAB      ; GET A LINE TO THE TEXP BUFFER.
8583  20B194             JSR    GETLIN

8586  C68F               DEC    TELN+3          ; REMOVE EOL.
8588  2028A1             JSR    TRAILB          ; PROCESS UNDERSCORE IF PRESENT.
858B  4CC785             JMP    :XA024

858E  A900       :XA021  LDA    #0
8590  A8                 TAY
8591  858A               STA    ACLN+2
```

```
8593  858E                  STA     TELN+2
8595  A920                  LDA     #' '            ; LEADING BLANK.
8597  918A                  STA     (ACLN),Y
8599  207497                JSR     XIN             ; GET KEY.
859C  918C                  STA     (TELN),Y
859E  C8                    INY
859F  848F                  STY     TELN+3
85A1  918A                  STA     (ACLN),Y
85A3  C8                    INY
85A4  4C1686                JMP     :XA030

85A7  C8          :XA022    INY                     ; YES -- SKIP OVER '='.
85A8  A586                  LDA     POINT           ; SAVE 'POINT'.
85AA  8D4905                STA     GNUMB
85AD  A587                  LDA     POINT+1
85AF  8D4A05                STA     GNUMB+1

85B2  20A7A0                JSR     TEXP            ; EVALUATE TEXT EXPRESSION.

85B5  A592                  LDA     EXEC            ; EXECUTE MODE?
85B7  F04A ^8603            BEQ     :XA009          ; NO.

85B9  84AC                  STY     XTEMP+1         ; YES -- RESTORE 'NP'.
85BB  A4AC                  LDY     XTEMP+1
85BD  AD4905                LDA     GNUMB           ; RESTORE 'POINT'.
85C0  8586                  STA     POINT
85C2  AD4A05                LDA     GNUMB+1
85C5  8587                  STA     POINT+1

                  ; *** EXTERNAL ENTRY POINT FROM 'XIN' ***

                  ;       EXPECTS: STATEMENT INDEX IN 'XTEMP+1'.
                  ;                TARGET ATOM TYPE IN 'XTEMP'.
                  ;                'POINT' OR 'NP' SETUP PER 'ATOM' CALL.
                  ;                'AXFLAG' SET PROPERLY.
                  ;                'SAVIT' CALLED IF STRING TARGET.
                  ;                STRING DATA IN 'TEXP'.
85C7              XAC024
85C7  20FF86      :XA024    JSR     RESIT           ; YES -- RESTORE NAME IF STRING TARGET.

85CA  A58E                  LDA     TELN+2          ; MOVE START INDEX.
85CC  858A                  STA     ACLN+2
85CE  AA                    TAX
85CF  A8                    TAY
85D0  AD4605                LDA     AXFLAG          ; ACCECPT LITERAL?
85D3  F012 ^85E7            BEQ     :XA24T          ; NO.

85D5  E48F        :XA24D    CPX     TELN+3          ; DONE?
85D7  F042 ^861B            BEQ     :XA031          ; YES.

85D9  BD008C                LDA     TEXBUF,X        ; NO -- GET NEXT CHAR.
85DC  E8                    INX

85DD  C0FE                  CPY     #ACCLNG         ; ACCEPT BUFFER FULL?
85DF  F03A ^861B            BEQ     :XA031          ; YES.

85E1  918A                  STA     (ACLN),Y
```

```
85E3  C8                      INY
85E4  4CD585                  JMP     :XA240

85E7  A920          :XA24T    LDA     #' '             ; INSERT LEADING BLANK.
85E9  D016 ^8601              BNE     :XA027           ; (BRA).

85EB  E48F          :XA025    CPX     TELN+3           ; DONE?
85ED  F027 ^8616              BEQ     :XA030           ; YES.

85EF  BD00BC                  LDA     TEXBUF,X         ; NO -- GET NEXT CHAR.
85F2  E8                      INX

85F3  C0FD          :XA026    CPY     #ACCLNG-1
85F5  F01F ^8616              BEQ     :XA030           ; ACCEPT BUFFER FULL.

                    ; CHARACTER CONVERSION HERE.

85F7  C961                    CMP     #'A'+$20         ; LOWER CASE ALPHA?
85F9  9006 ^8601              BCC     :XA027           ; NO.

85FB  C97B                    CMP     #'Z'+1+$20
85FD  B002 ^8601              BCS     :XA027           ; NO.

85FF  4920                    EOR     #$20             ; YES -- CONVERT TO UPPER CASE.

8601  9188          :XA027    STA     (ACLN),Y
8603  C8                      INY

8604  C920                    CMP     #' '             ; BLANK?
8606  D0E3 ^85EB              BNE     :XA025           ; NO.

8608  E48F          :XA028    CPX     TELN+3           ; YES -- SKIP MULTIPLES.
860A  F00F ^861B              BEQ     :XA031           ; END OF TEXT.

860C  BD00BC                  LDA     TEXBUF,X         ; GET NEXT CHARACTER.
860F  E8                      INX
8610  C920                    CMP     #' '             ; BLANK?
8612  D0DF ^85F3              BNE     :XA026           ; NO -- STORE IT.

8614  F0F2 ^8608              BEQ     :XA028           ; YES -- IGNORE IT (BRA).

8616  A920          :XA030    LDA     #' '             ; ADD TRAILING BLANK.
8618  9188                    STA     (ACLN),Y
861A  C8                      INY

861B  8488          :XA031    STY     ACLN+3           ; END INDEX.

861D  A5AB                    LDA     XTEMP            ; CHECK PARAMETER TYPE AGAIN.
861F  C901                    CMP     #NULL
8621  F02F ^8652              BEQ     :XA190           ; NONE -- ALL DONE.

8623  2984                    AND     #NVAR+EPTR
8625  D003 ^862A              BNE     :XA100           ; NUMERIC VARIABLE.

8627  4CA78B                  JMP     XCM300           ; STRING VARIABLE -- GO TO COMMON CODE & RET.

862A  A0FF          :XA100    LDY     #-1              ; CONVERT NUMBER TO BINARY REPRESENTATION.
```

```
862C  C8          :XA110  INY                      ; SCAN TO NUMBER OR EOL.
862D  B18C                LDA     (TELN),Y         ; GET A CHAR.
862F  C99B                CMP     #EOL             ; END OF LINE?
8631  F009 ^863C          BEQ     :XA120           ; YES -- DONE.

8633  C92D                CMP     #'-'             ; NO -- MINUS SIGN?
8635  F005 ^863C          BEQ     :XA120           ; YES -- DONE.

8637  20839E              JSR     CNUMBR           ; NO -- NUMERIC DIGIT?
863A  B0F0 ^862C          BCS     :XA110           ; NO -- KEEP SCANNING.

863C  A20C        :XA120  LDX     #TELN-DTAB       ; NOW CONVERT NUMBER WE FOUND.
863E  20B89D              JSR     ASCDEC

8641  A000                LDY     #0               ; MOVE VALUE TO VARIABLE.
8643  A588                LDA     NUMBER
8645  9186                STA     (POINT),Y
8647  A5AB                LDA     XTEMP            ; SEE IF POINTER TO BYTE.
8649  C980                CMP     #BPTR
864B  F005 ^8652          BEQ     :XA190           ; YES -- ALL DONE.

864D  C8                  INY
864E  A589                LDA     NUMBER+1
8650  9186                STA     (POINT),Y

8652  A4AC        :XA190  LDY     XTEMP+1          ; RESTORE LINE POINTER.
8654  A900                LDA     #0               ; SET CC FOR NORMAL EXIT.
8656  60                  RTS                      ; RETURN WITH CC SET.

                  ;
                  ; XACCX -- ACCEPT LITERAL COMMAND PROCESSOR.
                  ;
8657  A901        XACCX   LDA     #1
8659  8D4605              STA     AXFLAG
865C  A900                LDA     #0
865E  8D4705              STA     AKFLAG
8661  4C4B85              JMP     :XA001

                  ;
                  ; XACCK -- ACCEPT FROM KEYBOARD (SINGLE CHARACTER).
                  ;
8664  A901        XACCK   LDA     #1
8666  8D4705              STA     AKFLAG
8669  4C4B85              JMP     :XA001
```

```
866C                      PROC
                   ;
                   ; XMATCH -- MATCH COMMAND PROCESSOR
                   ;
866C  A900         XMATCH  LDA    #0               ; FORCE UPPER CASE ALPHA.
866E  F002 ^8672           BEQ    :XM005

8670  A901         XMATX   LDA    #1               ; LITERAL MATCH.

8672  8DDA05       :XM005  STA    LITMAT

8675  B180                 LDA    (INLN),Y         ; GET FIRST MATCH FIELD BYTE.
8677  C99B                 CMP    #EOL
8679  D003 ^867E           BNE    :XM010

867B  A902                 LDA    #IMPERR          ; NULL MATCH FIELD IS ERROR.

867D  60           :XM009  RTS                     ; RETURN WITH CC SET.

867E  20A7A0       :XM010  JSR    TEXP             ; EVALUATE TEXT EXPRESSION OPERAND.

8681  A592                 LDA    EXEC             ; EXECUTE MODE?
8683  F0F8 ^867D           BEQ    :XM009           ; NO -- DONE.

8685  A900                 LDA    #0               ; RESET MATCH FIELD NUMBER AND FLAG.
8687  85FE                 STA    MATCHF

8689  A92C                 LDA    #','              ; ',' IS DEFAULT MATCH FIELD DELIMITER.
868B  85E2                 STA    MFDEL

868D  84AC                 STY    XTEMP+1          ; SAVE INPUT INDEX.
868F  A58E                 LDA    TELN+2           ; CHECK FOR NULL RESULT.
8691  C58F                 CMP    TELN+3
8693  D003 ^8698           BNE    :XM011
8695  4C2387               JMP    :XM400           ; NULL PATTERN -- NO MATCH.

8698  ADDA05       :XM011  LDA    LITMAT           ; LITERAL MATCH?
869B  D012 ^86AF           BNE    :XM020           ; YES.

869D  A48E                 LDY    TELN+2           ; NO -- FORCE UPPER CASE ALPHA.

869F  B18C         :XM012  LDA    (TELN),Y         ; GET CHAR.
86A1  20919E               JSR    CLETTR           ; IS IT A LETTER?
86A4  B004 ^86AA           BCS    :XM015           ; NO.

86A6  29DF                 AND    #UC              ; YESS -- FORCE UPPER CASE.
86A8  918C                 STA    (TELN),Y

86AA  C8           :XM015  INY                     ; NEXT CHAR.
86AB  C48F                 CPY    TELN+3           ; DONE?
86AD  D0F0 ^869F           BNE    :XM012           ; NO.

                   ; THROUGHOUT THE MAIN LOOP THE X REGISTER WILL = ACCEPT START INDEX.

86AF  A68A         :XM020  LDX    ACLN+2           ; ACCEPT BUFFER START INDEX.
86B1  A48E                 LDY    TELN+2           ; SETUP MATCH PATTERN START INDEX.
```

```
86B3  B18C                     LDA    (TELN),Y        ; CHECK FOR ALTERNATE FIELD DELIMITER.
86B5  C97C                     CMP    #VBAR
86B7  D005 ^86BE               BNE    :XM050          ; NO ALTERNATE SPECIFIED.

86B9  85E2                     STA    MFDEL           ; SET ALTERNATE.
86BB  C8                       INY                    ; SKIP OVER VERTICAL BAR.
86BC  D00C ^86CA               BNE    :XM060          ; (BRA).

86BE  B18C          :XM050     LDA    (TELN),Y        ; GET 1ST CHAR OF OPERAND.
86C0  C91F                     CMP    #CRIGHT         ; RIGHT ARROW?
86C2  D00C ^86D0               BNE    :XM100          ; NO.

86C4  E8                       INX                    ; YES -- SKIP FIRST CHAR IN ACCEPT BUFFER.
86C5  C8                       INY                    ; SKIP OVER RIGHT ARROW TOO.

86C6  E48B                     CPX    ACLN+3          ; NULL ACCEPT BUFFER?
86C8  F059 ^8723               BEQ    :XM400          ; YES -- NO MATCH.

86CA  C48F          :XM060     CPY    TELN+3          ; NULL OPERAND?
86CC  F055 ^8723               BEQ    :XM400          ; YES.

86CE  D0EE ^86BE               BNE    :XM050          ; NO (BRA).

86D0  84A8          :XM100     STY    XTEMP           ; MATCH DATA INDEX (INNER LOOP).
86D2  84A2                     STY    TEMP+1          ; MATCH DATA INDEX (OUTER LOOP).

86D4  86A1                     STX    TEMP
86D6  868A                     STX    ACLN+2
86D8  E6FE                     INC    MATCHF          ; INCREMENT MATCH FIELD NUMBER.

86DA  A4A8          :XM120     LDY    XTEMP           ; SEE IF ALL OF PATTERN HAS MATCHED.
86DC  E6A8                     INC    XTEMP
86DE  C48F                     CPY    TELN+3
86E0  F037 ^8719               BEQ    :XM300          ; YES.

86E2  B18C                     LDA    (TELN),Y        ; NOT SURE.
86E4  C5E2                     CMP    MFDEL
86E6  F031 ^8719               BEQ    :XM300          ; YES.

86E8  A48A                     LDY    ACLN+2          ; NO -- MORE DATA TO MATCH?
86EA  E68A                     INC    ACLN+2
86EC  C48B                     CPY    ACLN+3
86EE  F004 ^86F4               BEQ    :XM140          ; NO -- AT END OF BUFFER.

86F0  D188                     CMP    (ACLN),Y        ; YES -- COMPARE DATA TO PATTERN.
86F2  F0E6 ^86DA               BEQ    :XM120          ; SO FAR SO GOOD.

86F4  A5A2          :XM140     LDA    TEMP+1          ; RESET MATCH PATTERN INDEX.
86F6  85A8                     STA    XTEMP
86F8  E6A1                     INC    TEMP            ; INCREMENT 'ACCBUF' INDEX.
86FA  A5A1                     LDA    TEMP
86FC  858A                     STA    ACLN+2
86FE  C58B                     CMP    ACLN+3
8700  D0D8 ^86DA               BNE    :XM120

8702  A4A2                     LDY    TEMP+1          ; INCREMENT 'TEXBUF' INDEX TO NEXT FIELD.
```

```
8704  B18C          :XM160  LDA     (TELN),Y
8706  C48F                  CPY     TELN+3          ; END OF MATCH PATTERN DATA?
8708  F009 ^8713            BEQ     :XM200          ; YES -- NO MATCH.

870A  C8                    INY
870B  C5E2                  CMP     MFDEL
870D  D0F5 ^8704            BNE     :XM160          ; KEEP SCANNING.

870F  C48F                  CPY     TELN+3          ; END OF MATCH STATEMENT?
8711  D0BD ^86D0            BNE     :XM100          ; NO.

8713  A900          :XM200  LDA     #0              ; NO MATCH -- RESET FLAG.
8715  85FE                  STA     MATCHF
8717  F00A ^8723            BEQ     :XM400          ; (BRA).

8719  A541          :XM300  LDA     TEMP            ; SAVE START & END INDICES TO MATCH FIELD ...
871B  8D3305                STA     MATCHX          ; ... FOR 'XMWSP'.
871E  A58A                  LDA     ACLN+2
8720  8D3405                STA     MATCHX+1

8723  A44C          :XM400  LDY     XTEMP+1         ; RESTORE INPUT LINE INDEX.

8725  A900                  LDA     #0              ; CLEAR LINE INDEX.
8727  858A                  STA     ACLN+2
8729  4C1B9F                JMP     SCNEOL          ; SCAN TO END OF INPUT LINE & RETURN.
```

```
8720                       PROC
                   ;
                   ; XMWSP -- MATCH WITH STRING PRODUCTION COMMAND PROCESSOR
                   ;
872C  207086       XMATSX  JSR     XMATX
872F  4C3587               JMP     :XM005

8732  206C86       XMWSP   JSR     XMATCH          ; FIRST DO ALL OF MATCH COMMAND.
8735  D035 ^876C   :XM005  BNE     :XM090          ; SYNTAX ERROR.

8737  A592                 LDA     EXEC            ; EXECUTE MODE?
8739  F031 ^876C           BEQ     :XM090          ; NO -- DONE (SYNTAX SAME AS MATCH).

873B  A5FE                 LDA     MATCHF          ; WAS MATCH SUCCESSFUL?
873D  F02D ^876C           BEQ     :XM090          ; NO -- ALL DONE.

873F  84AB                 STY     XTEMP
8741  A980                 LDA     #ATRSTR         ; 'STRING' ATTRIBUTE.
8743  8D6605               STA     ATRTYP
8746  8A                   TXA                     ; NOW SET $LEFT = DATA FROM ACCEPT START ...
8747  AC3305               LDY     MATCHX          ; ... TO START OF MATCH - 1.
874A  A200                 LDX     #LFTSTG-STAB
874C  206D87               JSR     MAKSTG
874F  D017 ^8768           BNE     :XM080          ; ERROR.

8751  AD3305               LDA     MATCHX          ; THEN SET $MATCH = DATA FROM MATCH.
8754  AC3405               LDY     MATCHX+1
8757  A205                 LDX     #MATSTG-STAB
8759  206D87               JSR     MAKSTG
875C  D00A ^8768           BNE     :XM080          ; ERROR.

875E  AD3405               LDA     MATCHX+1        ; THEN $RIGHT = DATA FROM MATCH +1 ...
8761  A48B                 LDY     ACLN+3          ; ... TO END.
8763  A206                 LDX     #RITSTG-STAB
8765  206D87               JSR     MAKSTG

8768  08           :XM080  PHP                     ; SAVE CC.
8769  A4AB                 LDY     XTEMP           ; RESTORE INDEX.
876B  28                   PLP

876C  60           :XM090  RTS                     ; RETURN WITH CC SET.


876D  85C4         MAKSTG  STA     DP+2            ; DEFINE DATA PORTION.
876F  84C5                 STY     DP+3
8771  A588                 LDA     ACLN
8773  85C2                 STA     DP
8775  A589                 LDA     ACLN+1
8777  85C3                 STA     DP+1

8779  BD8F87               LDA     STAB,X          ; DEFINE NAME PORTION.
877C  85C1                 STA     NP+3
877E  E8                   INX
877F  86C0                 STX     NP+2
8781  A98F                 LDA     # LOW STAB
8783  85BE                 STA     NP
```

```
8781  A98F                LDA     # LOW STAB
8783  858E                STA     NP
```

```
8785  A987                LDA     # HIGH STAB
8787  858F                STA     NP+1

8789  20AD9E              JSR     SETSVL          ; NAMED STRING VARIABLE LIST.

878C  4C0599              JMP     SINSRT          ; INSERT STRING & RETURN WITH CC SET.

      = 878F       STAB = *                        ; MATCH STRING NAME TABLE.

878F  054C454654  LFTSTG  DB   LSEND,'LEFT'
      = 0005      LSEND = *-STAB

8794  0B4D415443  MATSTG  DB   MSEND,'MATCH'
      = 000B      MSEND = *-STAB

879A  1152494748  RITSTG  DB   RSEND,'RIGHT'
      = 0011      RSEND = *-STAB
```

```
87A0                            PROC
                       ;
                       ; XNEWV -- NEW VARIABLES COMMAND PROCESSOR
                       ;
87A0  A592             XNEWV   LDA     EXEC                 ; EXECUTE MODE?
87A2  F012 ^87B6               BEQ     :XN090               ; SYNTAX SCAN ONLY.

87A4  A5B4                     LDA     S2H                  ; CLEAR MOD VARIABLES.
87A6  85B2                     STA     S2L
87A8  A5B5                     LDA     S2H+1
87AA  85B3                     STA     S2L+1

87AC  208198                   JSR     CLOSEM               ; CLOSE IOCBS 3 THROUGH 7.

87AF  A5FF                     LDA     RUN                  ; RUN MODE?
87B1  D003 ^87B6               BNE     :XN090               ; YES -- DON'T PRINT 'READY'.

87B3  202CB5                   JSR     RDYMES               ; NO -- PRINT 'READY'.

87B6                   XNE090
87B6  A900             :XN090  LDA     #0                   ; SET CC FOR EXIT.
87B8  60                       RTS                          ; RETURN WITH CC SET.

87B9                            PROC
                       ;
                       ; XNEW -- NEW PROGRAM PROCESSOR
                       ;
87B9  F0FB ^87B6       XNEW    BEQ     XNE090               ; SYNTAX SCAN ONLY.

87BB  20C087                   JSR     CLRPRG               ; CLEAR THE PROGRAM STORAGE AREA.
87BE  F0E0 ^87A0               BEQ     XNEWV                ; (BRA) NOW CLEAR THE VARIABLES ALSO.

87C0  A5AE             CLRPRG  LDA     S1L                  ; YES -- CLEAR PROGRAM STORAGE AREA.
87C2  85B0                     STA     S1H
87C4  A5AF                     LDA     S1L+1
87C6  85B1                     STA     S1H+1
87C8  A9FF                     LDA     #$FF                 ; NO CONTINUATION.
87CA  8D4305                   STA     NOCONT

87CD  A900                     LDA     #0
87CF  8D4D05                   STA     USTKP                ; CLEAR USE STACK.
87D2  60                       RTS                          ; RETURN WITH CC AND A = ZERO.


87D3                            PROC
                       ;
                       ; XCALL -- CALL MEMORY LOCATION PROCESSOR
                       ;
87D3  20D49F           XCALL   JSR     EXP                  ; ADDRESS SHOULD FOLLOW.

87D6  A592                     LDA     EXEC                 ; EXECUTE MODE?
87D8  F00B ^87E5               BEQ     :XC090               ; NO.

87DA  98                       TYA                          ; SAVE THE LINE INDEX FOR THE USER.
87DB  48                       PHA
```

```
87DA  98                    TYA                              ; SAVE THE LINE INDEX FOR THE USER.
87DB  48                    PHA
```

```
87DC  20E687                JSR     :XC100           ; "OFF WE GO, INTO THE WILD BLUE YONDER".

87DF  68                    PLA                      ; UNBELIEVEABLE, THE USER RETURNED.
87E0  A8                    TAY                      ; RESTORE THE LINE INDEX.
87E1  58                    CLI                      ; JUST IN CASE!
87E2  D8                    CLD                      ; DITTO.

87E3  A900                  LDA     #0               ; SET CC FOR EXIT.
87E5  60           :XC090   RTS                      ; RETURN WITH CC SET.

87E6  6C9300       :XC100   JMP     (EXPSTK)         ; TOO LATE TO CHANGE YOUR MIND.
```

```
87E9                        PROC
                    ;
                    ; XUSE -- USE COMMAND PROCESSOR
                    ;
87E9  F024 ^880F    XUSE    BEQ     XJMP          ; LET 'XJMP' PERFORM SYNTAX CHECK.

87EB  A5FF                  LDA     RUN           ; IF IMMEDIATE -- DON'T PUT ANYTHING IN STACK.
87ED  F01B ^880A            BEQ     :XU100

87EF  AE4D05                LDX     USTKP         ; USE STACK POINTER.
87F2  E030                  CPX     #USTKSZ
87F4  F011 ^8807            BEQ     :XU090        ; STACK FULL.

87F6  A584                  LDA     NXTLN         ; NEXT LINE ADDRESS TO USE STACK.
87F8  9D6B05                STA     USESTK,X
87FB  A585                  LDA     NXTLN+1
87FD  9D6C05                STA     USESTK+1,X
8800  E8                    INX
8801  E8                    INX
8802  8E4D05                STX     USTKP
8805  D00D ^8814            BNE     XJP005        ; REST OF COMMAND IS JUST LIKE 'J:' (BRA).

8807  A9FB          :XU090  LDA     #USOERR       ; STACK OVERFLOW ERROR.
8809  60                    RTS

880A  8D4D05        :XU100  STA     USTKP         ; CLEAR USE STACK.
880D  F005 ^8814            BEQ     XJP005        ; (BRA).
```

```
880F                        PROC
                   ;
                   ; XJMP -- JUMP COMMAND PROCESSOR
                   ;
880F  D003 ^8814  XJMP     BNE     XJP005          ; EXECUTE MODE.

8811  4CD39E               JMP     SCNLBL          ; SCAN OVER LABEL & RETURN.



                   ; *** EXTERNAL ENTRY POINT (FROM 'XJMPM' & 'XUSE') ***

8814  20139F      XJP005   JSR     SLB
8817  C8                   INY                     ; SKIP OVER '*'.
8818  84C4                 STY     DP+2            ; SETUP 'DP' TO POINT TO JUMP LABEL.
881A  20CB9E               JSR     SCECA           ; SCAN TO END OF LABEL.
881D  84C5                 STY     DP+3
881F  A580                 LDA     INLN
8821  85C2                 STA     DP
8823  A581                 LDA     INLN+1
8825  85C3                 STA     DP+1

8827  209F9E               JSR     STMLST          ; SETUP TO SCAN STATEMENT LIST.
882A  84AB                 STY     XTEMP           ; SAVE INPUT LINE POINTER.

882C  A23A        :XJ030   LDX     #LP-DTAB        ; CHECK FOR END OF STATEMENT LIST.
882E  20139A               JSR     SEND
8831  F040 ^8873           BEQ     :XJ200          ; END OF LIST -- LABEL NOT FOUND.

8833  A006                 LDY     #6              ; CHECK FOR PRESENCE OF LABEL.
8835  B1EA        :XJ032   LDA     (LP),Y
8837  C920                 CMP     #' '            ; BLANK?
8839  D003 ^883E           BNE     :XJ034          ; NO.
883B  C8                   INY                     ; SKIP LEADING BLANKS.
883C  D0F7 ^8835           BNE     :XJ032          ; (BRA).

883E  C92A        :XJ034   CMP     #'*'
8840  D029 ^886B           BNE     :XJ060          ; NO -- TRY NEXT STATEMENT.

8842  C8                   INY
8843  84C8                 STY     MP+2            ; YES -- SETUP 'MP' TO POINT TO STATEMENT LABEL.

8845  B1EA        :XJ040   LDA     (LP),Y          ; SCAN TO END OF LABEL.
8847  C8                   INY
8848  20B69E               JSR     CKECA           ; END OF ATOM (LABEL)?
884B  D0F8 ^8845           BNE     :XJ040          ; NO.

884D  88                   DEY
884E  84C9                 STY     MP+3

8850  A5EA                 LDA     LP              ; SETUP POINTERS FOR ...
8852  85C6                 STA     MP              ; ... 'SCOMP' CALL ...
8854  8584                 STA     NXTLN           ; ... & STATEMENT TO EXECUTE.
8856  A5EB                 LDA     LP+1
8858  85C7                 STA     MP+1
885A  8585                 STA     NXTLN+1
```

```
885C  205599            JSR     SCOMP            ; COMPARE LABELS.
885F  D00A ^886B        BNE     :XJ060           ; NO MATCH.

8861  A4AB              LDY     XTEMP            ; RESTORE INPUT LINE POINTER.
8863  84FF              STY     RUN              ; SET RUN MODE EVEN IF ALREADY SET.
8865  A900              LDA     #0
8867  8D4305            STA     NOCONT
886A  60                RTS                      ; RETURN WITH CC SET.

886B  A23A       :XJ060 LDX     #LP-DTAB         ; GET POINTER TO NEXT STATEMENT.
886D  20AA9A            JSR     SNXTI
8870  4C2C88            JMP     :XJ030

8873  A4C4       :XJ200 LDY     DP+2             ; RESTORE LINE INDEX.
8875  A90A              LDA     #UNDERR          ; UNDEFINED LABEL.
8877  60                RTS                      ; RETURN WITH CC SET.
```

```
8878                          PROC
                      ;
                      ; XJMPM -- JUMP ON MATCH RESULT COMMAND PROCESSOR
                      ;
8878  D00C ^8886      XJMPM   BNE     :XJ030          ; EXECUTE MODE.

887A  20D39E                  JSR     SCNLBL          ; SCAN OVER FIRST LABEL.
887D  D026 ^88A5              BNE     :XJ090          ; NOT EVEN ONE LABEL -- ERROR.

887F  20D39E      :XJ010      JSR     SCNLBL          ; SCAN OVER REMAINING LABELS.
8882  F0FB ^887F              BEQ     :XJ010

8884  D01D ^88A3              BNE     :XJ050          ; NORMAL RETURN.


8886  A5FE        :XJ030      LDA     MATCHF          ; WAS PREVIOUS MATCH SUCCESSFUL?
8888  F011 ^889B              BEQ     :XJ043          ; NO -- NO JUMP.

888A  AA                      TAX                     ; YES -- USE FIELD # AS LOOP COUNT.

888B  CA          :XJ040      DEX
888C  D010 ^889E              BNE     :XJ045          ; NOT THERE YET.

888E  20079F                  JSR     SKPSEP          ; PRE-VALIDATE NEXT LABEL.
8891  20F99E                  JSR     CHKTRM          ; END OF STATEMENT?
8894  F00F ^88A5              BEQ     :XJ090          ; YES -- O.K.

8896  201488                  JSR     XJP005          ; LET 'XJMP' DO THE DIRTY WORK.
8899  D00A ^88A5              BNE     :XJ090          ; ERROR.

889B  4C169F      :XJ043      JMP     SCNEOL          ; SCAN TO END OF STATEMENT & RETURN.

889E  20D39E      :XJ045      JSR     SCNLBL          ; SCAN OVER LABEL.
88A1  F0E8 ^888B              BEQ     :XJ040          ; THERE WAS ONE THERE.

88A3  A900        :XJ050      LDA     #0              ; TOO FEW LABELS IS O.K.

88A5  60          :XJ090      RTS                     ; RETURN WITH CC SET.
```

```
8846                        PROC
                    ;
                    ; XDUMP -- STRING & NUMERIC VARIABLE DUMP COMMAND PROCESSOR
                    ;
8846  20139F        XDUMP   JSR     SLB                 ; SKIP LEADING BLANKS.
8849  20F99E                JSR     CHKTRM              ; STATEMENT TERMINATOR?
884C  F00C ^888A            BEQ     :XD020              ; YES.

884E  C923                  CMP     #'#'
8880  F007 ^8889            BEQ     :XD010              ; NUMERIC VARIABLES ONLY.

8882  C924                  CMP     #'$'
8884  F003 ^8889            BEQ     :XD010              ; STRING VARIABLES ONLY.

8886  A902                  LDA     #IMPERR             ; IMPROPER OPERAND.

8888  60            :XD009  RTS                         ; RETURN WITH CC SET.

8889  C8            :XD010  INY

888A  85AB          :XD020  STA     XTEMP               ; SAVE OPERAND
888C  A592                  LDA     EXEC
888E  F0F8 ^8888            BEQ     :XD009              ; SYNTAX SCAN.

88C0  84AC                  STY     XTEMP+1             ; YES -- SAVE INPUT LINE INDEX.
88C2  A97D                  LDA     #CLEAR              ; CLEAR SCREEN.
88C4  208294                JSR     CHOT
88C7  CEFE02                DEC     CSPFLG              ; SET DISPLAY CONTROL CHARS FLAG.

                    ; DUMP ALL OF THE STRING VARIABLES

88CA  A5AB                  LDA     XTEMP               ; CHECK OPERAND.
88CC  C923                  CMP     #'#'                ; NUMERIC ONLY?
88CE  F013 ^88E3            BEQ     :XD050              ; YES.

88D0  A910                  LDA     #16                 ; PRODUCE STRING VARIABLE HEADER.
88D2  20FFB4                JSR     MESSOT

88D5  A960                  LDA     #ATRSTR             ; STRINGS.
88D7  8D6705                STA     DMPTYP
88DA  208689                JSR     DMPVAR

                    ; DUMP ALL OF THE NUMERIC VARIABLES.

88DD  A5AB                  LDA     XTEMP               ; CHECK OPERAND.
88DF  C924                  CMP     #'$'                ; STRING ONLY.
88E1  F00D ^88F0            BEQ     :XD060              ; YES.

88E3  A911          :XD050  LDA     #17                 ; NUMERIC VARIABLE HEADER.
88E5  20FFB4                JSR     MESSOT

88E8  A980                  LDA     #ATRNUM             ; NUMERIC.
88EA  8D6705                STA     DMPTYP
88ED  208689                JSR     DMPVAR

                    ; DUMP THE I/O'S.
```

```
88F0  A5AB         :XD060  LDA     XTEMP              ; CHECK OPERATOR.
88F2  A4AC                 LDY     XTEMP+1            ; RESTORE INDEX.
88F4  20F99E               JSR     CHKTRM             ; TERMINATOR?
88F7  D07D ^8976           BNE     :XD090             ; NO -- '$' OR '#'.

88F9  A923                 LDA     #35                ; I/O HEADER.
88FB  20FFB4               JSR     MESSOT

88FE  A920                 LDA     #ATRIO             ; I/O'S
8900  8D6705               STA     DMPTYP
8903  208689               JSR     DMPVAR

                   ; DUMP THE CONTENT OF THE STACK.

8906  A912                 LDA     #18                ; PRODUCE USE STACK HEADER.
8908  20FFB4               JSR     MESSOT

890B  AE4005               LDX     USTKP              ; STACK EMPTY?
890E  F01D ^892D           BEQ     :XD088             ; YES.

8910  20A29F       :XD087  JSR     SPACE              ; NO -- PRINT LINE #(S).
8913  BD6905               LDA     USESTK-2,X         ; GET POINTER TO STORED LINE.
8916  8586                 STA     POINT
8918  BD6A05               LDA     USESTK-1,X
891B  8587                 STA     POINT+1

891D  208C9F               JSR     GTLNNO             ; EXTRACT LINE NUMBER.

8920  86AD                 STX     XTEMP+2
8922  A25C                 LDX     #LINENO-DTAB       ; PRINT LINE NUMBER.
8924  20149E               JSR     DECASC
8927  A6AD                 LDX     XTEMP+2
8929  CA                   DEX
892A  CA                   DEX
892B  D0E3 ^8910           BNE     :XD087             ; MORE TO PRINT.

                   ; DUMP THE GRAPHICS PARAMETERS

892D  A913         :XD088  LDA     #19                ; PRODUCE GRAPHICS HEADER.
892F  20FFB4               JSR     MESSOT

8932  A958                 LDA     #'X'                ; X=FLOOR(<VALUE>).
8934  208294               JSR     CHOT
8937  208189               JSR     PRTEQS             ; '='.
893A  A26C                 LDX     #GX-DTAB
893C  20149E               JSR     DECASC
893F  209D9F               JSR     SPACES

8942  A959                 LDA     #'Y'                ; Y=FLOOR(<VALUE>).
8944  208294               JSR     CHOT
8947  208189               JSR     PRTEQS             ; '='.
894A  A26F                 LDX     #GY-DTAB
894C  20149E               JSR     DECASC
894F  209D9F               JSR     SPACES

8952  A914                 LDA     #20                ; THETA=<VALUE>.
8954  20FFB4               JSR     MESSOT
```

```
                    ATARI CAMAC Assembler Ver  1.0A  Page  83
 PILOT -- H.B. STEWART                                    D1:PILOT.

 8957  A27C              LDX     #THETA-DTAB
 8959  20149E            JSR     DECASC

                 ; REPORT ON FREE MEMORY

 895C  A915              LDA     #21              ; FREE MEMORY = <VALUE>.
 895E  20FF9A            JSR     MESSOT

 8961  A032              LDY     #S2L-DTAB        ; <VALUE> = 'S2L' - 'S1H' + 1.
 8963  20429D            JSR     DLOADA
 8966  A030              LDY     #S1H-DTAB
 8968  20619D            JSR     DSUBA
 896B  A901              LDA     #1
 896D  20049D            JSR     DADDS
 8970  20149E            JSR     DECASC           ; PRINT RESULT.
 8973  20989F            JSR     NEWLIN

 8976  20989F    :XD090  JSR     NEWLIN           ; BLANK LINE AFTER DUMP.
 8979  EEFE02            INC     DSPFLG           ; RESET DISPLAY CONTROL CHARS FLAG.
 897C  A4AC              LDY     XTEMP+1          ; DONE -- RESTORE INPUT LINE INDEX.
 897E  A900              LDA     #0               ; SET CC FOR EXIT.
 8980  60                RTS                      ; RETURN WITH CC SET.

 8981  A93D      PRTEQS  LDA     #'='             ; PRINT '=' ...
 8983  4C8294            JMP     CHOT             ; ... & RETURN.

 8986                    PROC
                 ; DMPVAR -- COMMON CODE FOR 'XDUMP'.
                 ;
                 ; CALLING SEQUENCE:
                 ;
                 ;       DMPTYP = ATTRIBUTE TYPE
                 ;
                 ;       JSR DMPVAR
                 ;
 8986  20AD9E    DMPVAR  JSR     SETSVL           ; POINT TO VARIABLE LIST.

 8989  A23A      :DM010  LDX     #LP-DTAB
 898B  20139A            JSR     SEND             ; END OF STRING STORAGE?
 898E  F060 ^89F0        BEQ     :DM090           ; YES -- DONE.

                 ; *S*   LDX     #LP-DTAB
 8990  20869A            JSR     SATTR            ; CORRECT TYPE?
 8993  CD6705            CMP     DMPTYP
 8996  D050 ^89E8        BNE     :DM080           ; NO.

 8998  A23B              LDX     #NUMBER-DTAB     ; MOVE POINTER TO 'NUMBER'.
 899A  A03A              LDY     #LP-DTAB
 899C  20459A            JSR     DMOVI

 899F  A924              LDA     #'$'
 89A1  2C6705            BIT     DMPTYP
 89A4  3004 ^89AA        BMI     :DM020           ; STRING.
 89A6  A923              LDA     #'#'
 89A8  5003 ^89AD        BVC     :DM030           ; I/O.
```

```
89AA  208294        :DM020  JSR     CHOT              ; PREFIX NAME FOR STRING, NUMERIC.

89AD  A002          :DM030  LDY     #2
89AF  20F189                JSR     PRTSFD            ; PRINT NAME.

89B2  208189                JSR     PRTEQS            ; SEPARATE NAME AND DATA WITH '='.

89B5  2C6705                BIT     DMPTYP
89B8  301E ^89D8            BMI     :DM040            ; STRING.

89BA  C8                    INY                       ; NUMERIC OR I/O.
89BB  B188                  LDA     (NUMBER),Y
89BD  5006 ^89C5            BVC     :DM032            ; I/O.
89BF  AA                    TAX                       ; NUMERIC.
89C0  C8                    INY
89C1  B188                  LDA     (NUMBER),Y
89C3  7007 ^89CC            BVS     :DM035            ; (BRA).

89C5  4A            :DM032  LSR     A                 ; IOCB = # * 16.
89C6  4A                    LSR     A
89C7  4A                    LSR     A
89C8  4A                    LSR     A
89C9  AA                    TAX
89CA  A900                  LDA     #0                ; MSB = 0.

89CC  8589          :DM035  STA     NUMBER+1          ; MSB.
89CE  8688                  STX     NUMBER            ; LSB.
89D0  A238                  LDX     #NUMBER-DTAB
89D2  20149E                JSR     DECASC            ; PRINT VALUE.
89D5  4CE589                JMP     :DM050

89D8  A927          :DM040  LDA     #SQUOTE           ; DELIMIT STRING DATA WITH '.
89DA  208294                JSR     CHOT
89DD  20F189                JSR     PRTSFD            ; PRINT STRING DATA.
89E0  A927                  LDA     #SQUOTE           ; CLOSING DELIMITER.
89E2  208294                JSR     CHOT

89E5  20989F        :DM050  JSR     NEWLIN

89E8  A23A          :DM080  LDX     #LP-DTAB          ; INCREMENT TO NEXT VARIABLE.
89EA  20AA9A                JSR     SNXTI
89ED  4C8989                JMP     :DM010

89F0  60            :DM090  RTS



89F1                        PROC
89F1  B188          PRTSFD  LDA     (NUMBER),Y        ; GET NAME/DATA LENGTH.
89F3  AA                    TAX
89F4  F012 ^8A08            BEQ     :PF090            ; DONE.

89F6  C8            :PF010  INY
89F7  D002 ^89FB            BNE     :PF020

89F9  E689                  INC     NUMBER+1          ; INDEX WRAPAROUND -- BUMP POINTER.
```

```
89FB  B188         :PF020  LDA    (NUMBER),Y       ; GET CHARACTER.
89FD  208294               JSR    CHOT
8A00  CA                   DEX                     ; DONE?
8A01  D0F3 ^89F6           BNE    :PF010           ; NO.

8A03  C8                   INY                     ; YES.
8A04  D002 ^8A08           BNE    :PF090

8A06  E6B9                 INC    NUMBER+1         ; INDEX WRAPAROUND -- BUMP POINTER.

8A08  60           :PF090  RTS
```

```
8A09                       PROC
                    ;
                    ; XPALET -- COLORS COMMAND PROCESSOR
                    ;

8A09  F013 ^8A1E   XPALET BEQ     :XP090          ; SYNTAX SCAN ONLY.

8A0B  A200                LDX     #0              ; SETUP TO SCAN COLOR NAME TABLE.

8A0D  204FA5       :XP010  JSR    PRNTCL          ; PRINT COLOR NAME FROM TABLE.
8A10  20909F              JSR     SPACES
8A13  E8                  INX                     ; SKIP OVER 'SB' ...
8A14  E8                  INX                     ; ... & COLOR VALUE.
8A15  E071                CPX     #PCTUP-PCTAB    ; END OF TABLE?
8A17  D0F4 ^8A0D          BNE     :XP010          ; NO -- CONTINUE.

8A19  20989F              JSR     NEWLIN
8A1C  A900                LDA     #0

8A1E  60           :XP090  RTS                    ; YES -- RETURN WITH CC SET.



                    ;
                    ; XCOLRS -- PS COMMAND PROCESSOR
                    ;

8A1F  F0FD ^8A1E   XCOLRS BEQ     :XP090          ; SYNTAX SCAN ONLY.

8A21  208B96              JSR     TSTMOD          ; GRAPHICS SPLIT SCREEN?
8A24  C904                CMP     #GRSS
8A26  D079 ^8AA1          BNE     :XC092          ; NO -- ERROR.

8A28  84A8                STY     XTEMP           ; SAVE Y REG.
8A2A  A927                LDA     #39             ; 'PENS: '.
8A2C  20FFB4              JSR     MESSOT
8A2F  A201                LDX     #1              ; SETUP TO EXAMINE COLOR ASSIGNS.

8A31  ECB905       :XC005  CPX    NCOLRS          ; END OF TABLE?
8A34  F002 ^8A38          BEQ     :XC010          ; NO.
8A36  B017 ^8A4F          BCS     :XC025          ; YES -- ALL DONE WITH PENS.

8A38  8A           :XC010  TXA                    ; GET PEN NUMBER.
8A39  4930                EOR     #'0'            ; CONVERT TO ASCII.
8A3B  208294              JSR     CHOT
8A3E  208189              JSR     PRTEQS          ; '='
8A41  ECBA05              CPX     NXTCLR          ; PEN ASSIGNED?
8A44  B003 ^8A49          BCS     :XC020          ; NO.

8A46  201AA5              JSR     PRCLNM          ; YES -- PRINT COLOR NAME.

8A49  209D9F       :XC020  JSR    SPACES
8A4C  E8                  INX                     ; NEXT PEN.
8A4D  D0E2 ^8A31          BNE     :XC005          ; (BRA).

8A4F  20989F       :XC025  JSR    NEWLIN
8A52  A928                LDA     #40             ; 'BACKGROUND: '.
```

```
8A54  20FFB4              JSR    MESSOT
8A57  A200                LDX    #0               ; BACKGROUND SLOT NUMBER.
8A59  201AA5              JSR    PRCLNM           ; PRINT COLOR NAME.

8A5C  A97F                LDA    #TAB
8A5E  208294              JSR    CHOT

8A61  A92A                LDA    #42              ; 'MODE: '
8A63  20FFB4              JSR    MESSOT
8A66  AD3705              LDA    GSMODE
8A69  85A9                STA    TEMP2+2
8A6B  A900                LDA    #0
8A6D  85AA                STA    TEMP2+3
8A6F  A229                LDX    #TEMP2+2-DTAB
8A71  20149E              JSR    DECASC           ; PRINT MODE NUMBER.
8A74  20989F              JSR    NEWLIN

8A77  A929                LDA    #41              ; 'TURTLE PEN: '.
8A79  20FFB4              JSR    MESSOT
8A7C  AD1305              LDA    PEN
8A7F  1004 ^8A85          BPL    :XC030           ; PEN DOWN.

8A81  A271                LDX    #PCTUP-PCTAB     ; 'UP'.
8A83  D002 ^8A87          BNE    :XC040           ; (BRA).

8A85  A275         :XC030 LDX    #PCTDN-PCTAB     ; 'DOWN'.

8A87  204FA5       :XC040 JSR    PRNTCL           ; PRINT 'UP' OR 'DOWN'.

8A8A  A92F                LDA    #'/'             ; '/'.
8A8C  208294              JSR    CHOT
8A8F  AD1305              LDA    PEN              ; NOW PRINT PEN NUMBER.
8A92  290F                AND    #$0F
8A94  4930                EOR    #'0'             ; CONVERT TO ASCII.
8A96  208294              JSR    CHOT
8A99  20989F              JSR    NEWLIN

8A9C  A4AE                LDY    XTEMP
8A9E  A900                LDA    #0               ; SET CC FOR EXIT.

8AA0  60           :XC090 RTS

8AA1  A983         :XC092 LDA    #NRCERR          ; COMMAND ONLY VALID IN GRSS.
8AA3  60                  RTS


8AA4                      PROC

                   ; XENVIR -- ES COMMAND, TURTLE ENVIRONMENT STATUS

8AA4  F077 ^8B1D   XENVIR BEQ    :XE090           ; SYNTAX SCAN ONLY.

8AA6  20B896              JSR    TSTMOD           ; GRAPHICS SPLIT SCREEN?
8AA9  C904                CMP    #GRSS
8AAB  D071 ^8B1E          BNE    :XE092           ; NO -- ERROR.

8AAD  84AE                STY    XTEMP            ; SAVE Y REGISTER.
```

```
8AAF  A92B                    LDA     #43             ; 'EDGE:'.
8AB1  20FFB4                  JSR     MESSOT
8AB4  A200                    LDX     #0

8AB6  86AC          :XE010    STX     XTEMP+1

8AB8  BD0080        :XE020    LDA     EDGTAB,X        ; SCAN TO NAME DELIMITER.
8ABB  E8                      INX
8ABC  C980                    CMP     #$80
8ABE  D0F8 ^8AB8              BNE     :XE020

8AC0  BD0080                  LDA     EDGTAB,X        ; SEE IF THIS IS THE RULE IN EFFECT.
8AC3  E8                      INX
8AC4  CD5E05                  CMP     EDGRUL
8AC7  D0ED ^8AB6              BNE     :XE010          ; NO -- SCAN TO NEXT NAME.

8AC9  A6AC                    LDX     XTEMP+1         ; YES -- BACKUP TO NAME TEXT.

8ACB  BD0080        :XE030    LDA     EDGTAB,X        ; GET A CHARACTER.
8ACE  3006 ^8AD6              BMI     :XE040          ; DELIMITER.

8AD0  208294                  JSR     CHOT            ; OUTPUT CHAR.
8AD3  E8                      INX
8AD4  D0F5 ^8ACB              BNE     :XE030          ; (BRA).

8AD6  A97F          :XE040    LDA     #TAB
8AD8  208294                  JSR     CHOT
8ADB  A92C                    LDA     #44             ; 'SPEED: '.
8ADD  20FFB4                  JSR     MESSOT
8AE0  AD5D05                  LDA     SPEED
8AE3  85A7                    STA     TEMP2
8AE5  A900                    LDA     #0
8AE7  85A8                    STA     TEMP2+1
8AE9  A227                    LDX     #TEMP2-DTAB
8AEB  20149E                  JSR     DECASC
8AEE  20989F                  JSR     NEWLIN

8AF1  A92D                    LDA     #45             ; 'WALLS: '.
8AF3  20FFB4                  JSR     MESSOT
8AF6  18                      CLC
8AF7  A001                    LDY     #1

8AF9  6ECE05        :XE050    ROR     WALLS+1
8AFC  6ECD05                  ROR     WALLS
8AFF  900F ^8B10              BCC     :XE060          ; NOT A WALL SELECT.

8B01  84A7                    STY     TEMP2           ; A WALL SELECT, PRINT POSITION #.
8B03  A900                    LDA     #0
8B05  85A8                    STA     TEMP2+1
8B07  A227                    LDX     #TEMP2-DTAB
8B09  20149E                  JSR     DECASC
8B0C  20A29F                  JSR     SPACE
8B0F  38                      SEC

8B10  C8            :XE060    INY
8B11  98                      TYA
8B12  4912                    EOR     #17+1           ; COMPARE WITHOUT ALTERING THE CARRY.
```

```
8B14  D0E3 ^8AF9            BNE     :XE050

8B16  20989F                JSR     NEWLIN
8B19  A4A0                  LDY     XTEMP
8B1B  A900                  LDA     #0

8B1D  60           :XE090   RTS

8B1E  A983         :XE092   LDA     #NRCERR         ; COMMAND VALID ONLY IN GRSS.
8B20  60                    RTS

8B21                        PROC
8B21  A20A         XTV      LDX     #ONOFFX         ; CHECK FOR 'ON' OR 'OFF'.
8B23  20A87C                JSR     SBCMAT
8B26  D014 ^8B3C            BNE     :XT090          ; ERROR.

8B28  A592                  LDA     EXEC            ; EXECUTE MODE?
8B2A  F010 ^8B3C            BEQ     :XT090          ; NO.

8B2C  8A                    TXA                     ; ON OR OFF?
8B2D  F00E ^8B3D            BEQ     :XT100          ; OFF.

8B2F  ADE005                LDA     DMASAV          ; ON -- TV OFF NOW?
8B32  F008 ^8B3C            BEQ     :XT090          ; NO.

8B34  8D2F02                STA     DMACT           ; YES -- RESTORE PRIOR STATE.
8B37  A900                  LDA     #0
8B39  8DE005                STA     DMASAV

8B3C  60           :XT090   RTS

8B3D  AD2F02       :XT100   LDA     DMACT           ; OFF -- IS TV ON NOW?
8B40  F0F4 ^8B3C            BEQ     :XT090          ; NO.

8B42  8DE005                STA     DMASAV          ; YES -- SAVE STATE.
8B45  A900                  LDA     #0              ; DMA OFF.
8B47  8D2F02                STA     DMACT
8B4A  60                    RTS
```

```
8848                         PROC
                    ;
                    ; COMPUTE COMMAND PROCESSOR
                    ;
884B  206E81        XCMPUT   JSR     ATOM            ; CHECK FOR TARGET VARIABLE.
884E  D06A ^88BA             BNE     :XC900          ; INVALID ATOM.

8850  C904                   CMP     #NVAR
8852  F00B ^885F             BEQ     :XC100          ; NUMERIC ASSIGNMENT.

8854  C980                   CMP     #BPTR           ; BYTE POINTER?
8856  F007 ^885F             BEQ     :XC100          ; YES -- SAME AS NUMERIC VARIABLE.

8858  2918                   AND     #SVAR+USVAR     ; STRING ASSIGNMENT?
885A  D036 ^8892             BNE     :XC200          ; YES.

885C  A902          :XC092   LDA     #IMPERR         ; NO -- ERROR.
885E  60                     RTS

                    ; ARITHMETIC ASSIGNMENT

885F  85AB          :XC100   STA     XTEMP           ; SAVE TARGET TYPE.

8861  20009F                 JSR     CHKEQS          ; CHECK FOR ASSIGNMENT OPERATOR NEXT.
8864  D0F6 ^885C             BNE     :XC092          ; ASSIGNMENT SYNTAX ERROR.

8866  A5B6                   LDA     POINT           ; SAVE TARGET ADDRESS.
8868  48                     PHA
8869  A5B7                   LDA     POINT+1
886B  48                     PHA

886C  C8                     INY                     ; PREPARE TO EVALUATE EXPRESSION.
886D  20C49F                 JSR     EXP             ; EVALUATE EXPRESSION.

8870  68                     PLA                     ; RESTORE TARGET ADDRESS.
8871  85B7                   STA     POINT+1
8873  68                     PLA
8874  85B6                   STA     POINT

8876  A592                   LDA     EXEC            ; EXECUTE MODE?
8878  F040 ^88BA             BEQ     :XC900          ; NO.

887A  84AC                   STY     XTEMP+1         ; SAVE LINE INDEX.
887C  A000                   LDY     #0              ; STORE RESULT TO TARGET.
887E  A593                   LDA     EXPSTK
8880  91B6                   STA     (POINT),Y
8882  A5AB                   LDA     XTEMP           ; SEE IF TARGET IS POINTER TO BYTE.
8884  C980                   CMP     #BPTR
8886  F005 ^888D             BEQ     :XC120          ; YES -- ALL DONE.

8888  C8                     INY                     ; NO -- NOW MSB.
8889  A594                   LDA     EXPSTK+1
888B  91B6                   STA     (POINT),Y

888D  A4AC          :XC120   LDY     XTEMP+1         ; RESTORE LINE INDEX.
888F  A900                   LDA     #0              ; COMPUTE WAS A SUCCESS.
```

```
8891  60                     RTS

                  ; STRING ASSIGNMENT

8892  20D09F      :XC200  JSR    CHKEQS           ; ASSIGNMENT OPERATOR?
8895  D0C5 ^885C          BNE    :XC092           ; NO -- ERROR.

8897  20C188              JSR    SAVIT2           ; SAVE 'NP' TO 'MP' TEMPORARILY.

889A  C8                  INY                     ; SKIP OVER '='.
889B  20A7A0              JSR    TEXP             ; EVALUATE TEXT EXPRESSION.

889E  A592                LDA    EXEC             ; EXECUTE MODE?
88A0  F018 ^88BA          BEQ    :XC900           ; NO -- DON'T DO ASSIGNMENT.

88A2  84AC                STY    XTEMP+1
88A4  20058C              JSR    RESIT2           ; RESTORE 'NP' FROM 'MP'.

                  ; *** EXTERNAL ENTRY POINT FROM 'XACCPT' ***

88A7  A242        XCM300  LDX    #DP-DTAB         ; MOVE 'TELN' TO 'DP'.
88A9  A00C                LDY    #TELN-DTAB
88AB  20359A              JSR    PMOVE

88AE  A980                LDA    #ATRSTR          ; 'STRING' ATTRIBUTE.
88B0  8D6605              STA    ATRTYP
88B3  200599              JSR    SINSRT           ; INSERT STRING.
88B6  08                  PHP
88B7  A4AC                LDY    XTEMP+1
88B9  28                  PLP

88BA  60          :XC900  RTS                     ; RETURN WITH CC SET.



88BB                      PROC
88BB  A5AB        SAVIT   LDA    XTEMP            ; STRING TARGET?
88BD  2918                AND    #SVAR+USVAR
88BF  F036 ^88F7          BEQ    :SV090           ; NO.

88C1  A592        SAVIT2  LDA    EXEC             ; EXECUTE MODE?
88C3  F032 ^88F7          BEQ    :SV090           ; NO.

88C5  98                  TYA                     ; SAVE Y REGISTER.
88C6  48                  PHA

88C7  A97B                LDA    # LOW NAMBUF     ; NAME SAVE AREA.
88C9  85C6                STA    MP
88CB  A98E                LDA    # HIGH NAMBUF
88CD  85C7                STA    MP+1
88CF  A002                LDY    #2               ; TARGET STRING START INDEX.
88D1  8C4205              STY    NAMLNG

88D4  A4C0                LDY    NP+2             ; SOURCE STRING START INDEX.
88D6  C4C1                CPY    NP+3             ; NULL SOURCE?
88D8  F01E ^88F8          BEQ    :SV100           ; YES -- ERROR.
```

```
8BDA  C4C1        :SV010  CPY     NP+3            ; END OF STRING?
8BDC  F017 ^8BF5          BEQ     :SV080          ; YES.

8BDE  B1BE                LDA     (NP),Y          ; NO -- GET A CHAR.
8BE0  C8                  INY
8BE1  84A1                STY     TEMP
8BE3  20B69E              JSR     CKEOA           ; END OF ATOM (STRING NAME)?
8BE6  F010 ^8BF8          BEQ     :SV100          ; YES -- ERROR.

8BE8  AC4205              LDY     NAMLNG          ; STORE A CHAR.
8BEB  91C6                STA     (MP),Y
8BED  C8                  INY
8BEE  8C4205              STY     NAMLNG

8BF1  A4A1                LDY     TEMP
8BF3  D0E5 ^8BDA          BNE     :SV010          ; TRY AGAIN (BRA).

8BF5  68          :SV080  PLA                     ; RESTORE Y REGISTER.
8BF6  A8                  TAY

8BF7  60          :SV090  RTS

8BF8  68          :SV100  PLA                     ; RESTORE Y REGISTER.
8BF9  A8                  TAY
8BFA  A982                LDA     #ATMERR+NS      ; INVALID STRING NAME.

8BFC  4C3A7A      :SV190  JMP     PSTOP           ; ABORT COMMAND.


8BFF                      PROC
8BFF  A5AB        RESIT   LDA     XTEMP           ; STRING TARGET?
8C01  2918                AND     # SVAR+USVAR
8C03  F011 ^8C16          BEQ     :RS090          ; NO.

8C05  A97B        RESIT2  LDA     # LOW NAMBUF
8C07  85BE                STA     NP
8C09  A98E                LDA     # HIGH NAMBUF
8C0B  85BF                STA     NP+1
8C0D  A902                LDA     #2
8C0F  85C0                STA     NP+2
8C11  AD4205              LDA     NAMLNG
8C14  85C1                STA     NP+3

8C16  60          :RS090  RTS
```

```
8C17                      PROC
                     ;
                     ; XGRAPH -- GRAPHICS COMMAND PROCESSOR
                     ;
8C17  A592           XGRAPH  LDA     EXFC             ; EXECUTE MODE?
8C19  F00F ^8C2A             BEQ     :XG020           ; NO.

8C1B  AD1405                 LDA     GRFLAG           ; YES -- GRAPHICS SCREEN OPEN?
8C1E  D00A ^8C2A             BNE     :XG020           ; YES.

8C20  2060AF                 JSR     GPINIT           ; INITIALIZE GRAPHICS PARAMETERS.
8C23  844B                   STY     XTEMP
8C25  201095                 JSR     GSOPEN           ; OPEN GRAPHICS SCREEN.
8C28  A44B                   LDY     XTEMP

8C2A  20398C         :XG020  JSR     GCOMND           ; PROCESS ONE GRAPHICS SUB-COMMAND.

8C2D  20139F                 JSR     SLB              ; SEE IF MULTIPLES.
8C30  C8                     INY
8C31  C93B                   CMP     #';'
8C33  F0E2 ^8C17             BEQ     XGRAPH           ; YES.

8C35  88                     DEY                      ; NO -- ALL DONE.
8C36  A900                   LDA     #0               ; CLEAR CC FOR NORMAL EXIT.
8C38  60                     RTS                      ; RETURN WITH CC SET.


8C39                         PROC
                     ; 'GCOMND' PROCESS ONE GRAPHICS SUB-COMMAND OR NESTED GROUP.

8C39  20139F         GCOMND  JSR     SLB              ; SKIP LEADING BLANKS.
8C3C  C928                   CMP     #'('              ; CHECK FOR GROUPING WITH '(' & ')'.
8C3E  F022 ^8C62             BEQ     :GC100

8C40  206E81                 JSR     ATOM             ; CHECK ATOM TYPE.
8C43  D01A ^8C5F             BNE     :GC090           ; ATOM ERROR.

8C45  2986                   AND     #NUM+NVAR+BPTR   ; IF NUMERIC, THEN TREAT AS ITERATION COUNT.
8C47  D02A ^8C73             BNE     :GC200           ; YEP.

8C49  A204                   LDX     #GTABX           ; NO -- ASSUME ITS A SUB-COMMAND.
8C4B  20AB7C                 JSR     SBCMAT
8C4E  D00F ^8C5F             BNE     :GC090           ; NO -- ERROR.

8C50  BDAA80                 LDA     SBDTAB,X         ; SETUP ADDRESS OF G-ROUTINE.
8C53  8D0B05                 STA     GJUMP+1
8C56  BDAB80                 LDA     SBDTAB+1,X
8C59  8D0C05                 STA     GJUMP+2

8C5C  4C0A05                 JMP     GJUMP            ; GO TO G-ROUTINE & RETURN.

8C5F  4C3A7A         :GC090  JMP     FSTOP            ; FATAL ERROR -- STOP EXECUTION.

                     ; THIS SECTION HANDLES NESTED GROUPS.

8C62  C8             :GC100  INY                      ; SKIP OVER '('.
8C63  20178C                 JSR     XGRAPH           ; PROCESS ONE SUB-COMMAND OR NESTED GROUP.
8C66  20139F              JSR     SLB
8C69  C8                  INY
8C6A  C929                CMP     #')'             ; MATCHING PAREN?
8C6C  F04A ^8CB8          BEQ     :GC390           ; YES -- O.K.

8C6E  88                  DEY                      ; NO -- ERROR.
8C6F  A902                LDA     #NSTERR
8C71  D0EC ^8C5F          BNE     :GC090           ; (BRA).

                    ; THIS SECTION HANDLES ITERATIONS

                    ;
                    ; *** EXTERNAL ENTRY POINT ***
                    ;
8C73                GITER

8C73  A592          :GC200 LDA    EXEC             ; EXECUTE MODE?
8C75  F034 ^8CAB          BEQ     :GC300           ; NO -- SYNTAX SCAN ONLY.

8C77  A5B8                LDA     NUMBER           ; SEE IF ZERO ITERATIONS.
8C79  05B9                ORA     NUMBER+1
8C7B  F02E ^8CAB          BEQ     :GC300           ; YES -- SCAN OVER ITERATION BODY.

8C7D  A5DE                LDA     LS               ; NO -- SAVE COUNTER ('LS') ...
8C7F  48                  PHA
8C80  A5DF                LDA     LS+1
8C82  48                  PHA
8C83  98                  TYA                      ; ... & LINE INDEX.
8C84  48                  PHA

8C85  A5B8                LDA     NUMBER           ; GET LOOP COUNT TO 'LS'.
8C87  85DE                STA     LS
8C89  A5B9                LDA     NUMBER+1
8C8B  85DF                STA     LS+1

8C8D  20398C        :GC220 JSR    GCOMND           ; PROCESS ONE COMMAND.

8C90  A25E                LDX     #LS-DTAB         ; DECREMENT ITERATION COUNT.
8C92  20129D              JSR     DDCRI
8C95  A5DE                LDA     LS               ; CHECK FOR RESULT = 0.
8C97  05DF                ORA     LS+1
8C99  F008 ^8CA3          BEQ     :GC240           ; DONE.

8C9B  207E9F              JSR     ABRTCK           ; CHECK FOR OPERATOR ABORT.
8C9E  68                  PLA                      ; NOT DONE -- RESTORE SCAN INDEX.
8C9F  48                  PHA
8CA0  A8                  TAY
8CA1  D0EA ^8C8D          BNE     :GC220           ; (BRA) EXECUTE BODY AGAIN.

8CA3  68            :GC240 PLA                     ; THROW AWAY STARTING INDEX.
8CA4  68                  PLA                      ; RESTORE 'LS'.
8CA5  85DF                STA     LS+1
8CA7  68                  PLA
8CA8  85DE                STA     LS
8CAA  60                  RTS
```

```
                        ; THIS SECTION SYNTAX SCANS THE BODY OF AN ITERATION.

8CAB  A592              :GC300  LDA    EXEC              ; SAVE CURRENT VALUE.
8CAD  48                        PHA
8CAE  A900                      LDA    #0                ; SETUP FOR SCAN ONLY
8CB0  8592                      STA    EXEC

8CB2  20398C                    JSR    GCOMND            ; *** RECURSIVE CALL ***

8CB5  68                        PLA                      ; RESTORE MODE.
8CB6  8592                      STA    EXEC

8CB8  60                :GC390  RTS                      ; RETURN WITH CC SET.


8CB9                            PROC
                        ;
                        ; GREPT -- 'REPEAT' GRAPHICS SUBCOMMAND
                        ;
8CB9  206E81            GREPT   JSR    ATOM              ; REPEAT COUNT MUST FOLLOW.
8CBC  D009 ^8CC7                BNE    :GR090            ; ERROR.

8CBE  2986                      AND    #NUM+NVAR+BPTR    ; NUMERIC DATA?
8CC0  F003 ^8CC5                BEQ    :GR088            ; NO -- ERROR.

8CC2  4C738C                    JMP    GITER             ; YES -- PROCESS REPEAT LOGIC.

8CC5  A902              :GR088  LDA    #IMPERR

8CC7  4C3A7A            :GR090  JMP    PSTOP
```

```
8CCA                        PROC
                    ;
                    ; XSOUND -- SOUND COMMAND PROCESSOR
                    ;
8CCA  A208          XSOUND  LDX     #AUREGS*2         ; SETUP INDEX TO # OF REGS.

8CCC  20079F        :XS010  JSR     SKPSEP            ; SKIP SEPARATORS & GET CHAR.
8CCF  20F99E                JSR     CHKTRM            ; TERMINATOR?
8CD2  F052 ^8D26            BEQ     :XS080            ; YES -- ALL DONE.

8CD4  C928                  CMP     #'('              ; LEFT PAREN?
8CD6  F04B ^8D23            BEQ     :XS050            ; YES -- START OF NOTE LIST.

8CD8  C929                  CMP     #')'              ; RIGHT PAREN?
8CDA  F04A ^8D26            BEQ     :XS080            ; YES -- END OF NOTE LIST.

8CDC  C93D                  CMP     #'='              ; EQUAL SIGN?
8CDE  F022 ^8D02            BEQ     :XS020            ; YES -- NO CHANGE FOR VOICE.

8CE0  C92B                  CMP     #'+'              ; PLUS SIGN?
8CE2  F021 ^8D05            BEQ     :XS030            ; YES -- INCREMENT NOTE.

8CE4  C92D                  CMP     #'-'              ; MINUS SIGN.
8CE6  F02C ^8D14            BEQ     :XS040            ; YES -- DECREMENT NOTE.

8CE8  20578D                JSR     GTNOTE            ; GET NUMERIC VALUE.
8CEB  D064 ^8D51            BNE     :XS090            ; ERROR.

8CED  A5B8                  LDA     NUMBER            ; NOTE := NUM.

8CEF  85AB          :XS015  STA     XTEMP             ; SAVE NOTE #
8CF1  A592                  LDA     EXEC              ; EXECUTE MODE?
8CF3  F007 ^8CFC            BEQ     :XS017            ; NO.

8CF5  A5AB                  LDA     XTEMP             ; YES.
8CF7  0980                  ORA     #$80              ; SET BIT FOR NOT A POINTER.
8CF9  9D1405                STA     AUDIOR-1,X

8CFC  CA            :XS017  DEX                       ; MORE OPERANDS ALLOWED?
8CFD  CA                    DEX
8CFE  D0CC ^8CCC            BNE     :XS010            ; YES.

8D00  F024 ^8D26            BEQ     :XS080            ; NO -- SEE IF DURATION (BRA).

8D02  C8            :XS020  INY
8D03  D0F7 ^8CFC            BNE     :XS017            ; (BRA).

8D05  C8            :XS030  INY
8D06  20578D                JSR     GTNOTE            ; GET INCREMENT VALUE.
8D09  D046 ^8D51            BNE     :XS090            ; ERROR.

8D0B  BD1405                LDA     AUDIOR-1,X        ; NOTE :=NOTE + NUM.
8D0E  18                    CLC
8D0F  65B8                  ADC     NUMBER
8D11  4CEF8C                JMP     :XS015

8D14  C8            :XS040  INY
```

```
               ATARI CAMAC Assembler Ver  1.0A  Page  97
PILOT -- H.B. STEWART                                   D1:PILOT.

8D15  20578D                JSR   GTNOTE          ; GET DECREMENT VALUE.
8D18  D037 ^8D51            BNE   :XS090          ; ERROR.

8D1A  BD1405                LDA   AUDIOR-1,X      ; NOTE := NOTE - NUM.
8D1D  38                    SEC
8D1E  E528                  SBC   NUMBER
8D20  4CEF8C                JMP   :XS015

8D23  C8          :XS050    INY                   ; SKIP OVER LEFT PAREN.
8D24  D0A6 ^8CCC            BNE   :XS010          ; (BRA).


8D26  A592        :XS080    LDA   EXEC            ; EXECUTE MODE?
8D28  F01A ^8D44            BEQ   :XS084          ; NO.

8D2A  E000                  CPX   #0
8D2C  F00F ^8D3D            BEQ   :XS083

8D2E  A900        :XS082    LDA   #0              ; CLEAR UNSPECIFIED VOICES.
8D30  9D1405                STA   AUDIOR-1,X
8D33  9DFED1                STA   AUDF1-2,X       ; CLEAR SOUND REGISTERS.
8D36  9DFFD1                STA   AUDC1-2,X
8D39  CA                    DEX
8D3A  CA                    DEX
8D3B  D0F1 ^8D2E            BNE   :XS082

8D3D  84AB        :XS083    STY   XTEMP
8D3F  204FB4                JSR   TONES
8D42  A4AB                  LDY   XTEMP

8D44  B180        :XS084    LDA   (INLN),Y        ; DURATION FOLLOWING?
8D46  C929                  CMP   #')'
8D48  D004 ^8D4E            BNE   :XS088          ; NO.

8D4A  C8                    INY                   ; YES -- SKIP OVER LEFT PAREN.
8D4B  4C548F                JMP   XWAIT           ; PROCESS DURATION AS A PAUSE.

8D4E  A900        :XS088    LDA   #0              ; RETURN WITH CC SET.
8D50  60                    RTS

8D51  20B49F      :XS090    JSR   AUDCLR          ; CLEAR ALL SOUND REGS.

8D54  A902        XIN080    LDA   #IMPERR

8D56  60          XIN090    RTS                   ; RETURN WITH CC SET.


8D57                        PROC
8D57  86AB        GTNOTE    STX   XTEMP           ; SAVE X REGISTER.
8D59  206E81                JSR   ATON            ; GET OPERAND.
8D5C  D008 ^8D66            BNE   :GN090          ; ERROR.

8D5E  2956                  AND   #NUM+NVAR+BPTR
8D60  F005 ^8D67            BEQ   :GN092          ; ERROR.

8D62  A6AB                  LDX   XTEMP           ; RESTORE X REGISTER.
```

```
8062  A6A9             LDX     XTEMP           ; RESTORE X REGISTER.
```

```
8064  A900             LDA     #0

8066  60        :GN090 RTS

8067  A902      :GN092 LDA     #IMPERR
8069  60               RTS
```

```
8D54                      PROC
                   ;
                   ; XIN -- READ COMMAND PROCESSOR
                   ;
8D6A  A904         XIN    LDA    #OREAD             ; READ DIRECTION.
8D6C  20FD97              JSR    SCNDEV             ; CONVERT DEVICE SPEC TO IOCB INDEX.
8D6F  D0E5 ^8D56          BNE    XIN090             ; ERROR.

8D71  86AD                STX    XTEMP+2            ; SAVE IOCB INDEX.
8D73  20079F              JSR    SKPSEP             ; SKIP OVER SEPARATOR.
8D76  206E81              JSR    ATOM               ; FIND TYPE OF VARIABLE.
8D79  D0DB ^8D56          BNE    XIN090             ; ERROR.

8D7B  85AB                STA    XTEMP              ; SAVE ATOM TYPE.
8D7D  299D                AND    #SVAR+USVAR+NVAR+NULL+BPTR ; VALID TYPE?
8D7F  F0D3 ^8D54          BEQ    XIN080             ; NO.

8D81  A592                LDA    EXEC               ; EXECUTE MODE?
8D83  F0D1 ^8D56          BEQ    XIN090             ; NO.

8D85  84AC                STY    XTEMP+1            ; SAVE LINE INDEX.
8D87  A6AD                LDX    XTEMP+2            ; GET IOCB INDEX.
8D89  A000                LDY    #0                 ; INIT INDEX TO ACCEPT BUFFER.
8D8B  848E                STY    TELN+2

8D8D  AD2005              LDA    OPNBUF             ; SEE IF READING FROM TEXT SCREEN.
8D90  C945                CMP    #'E'
8D92  D004 ^8D98          BNE    :XI030             ; NO.

8D94  98                  TYA                       ; YES -- ENABLE CURSOR (Y = 0).
8D95  20A79F              JSR    CRSNOP             ; MAKE IT APPEAR.

8D98  205397     :XI030   JSR    DIN                ; GET A CHARACTER FROM DEVICE.
8D9B  C99B                CMP    #EOL               ; END OF LINE?
8D9D  F00E ^8DAD          BEQ    :XI040             ; YES -- DONE.

8D9F  918C                STA    (TELN),Y
8DA1  C8                  INY
8DA2  C0FE                CPY    #TEXLNG            ; BUFFER FULL?
8DA4  D0F2 ^8D98          BNE    :XI030             ; NO.

8DA6  205397     :XI035   JSR    DIN                ; YES -- FLUSH TO EOL.
8DA9  C99B                CMP    #EOL
8DAB  D0F9 ^8DA6          BNE    :XI035

8DAD  848F       :XI040   STY    TELN+3             ; SAVE STRING END INDEX.
8DAF  AD2005              LDA    OPNBUF             ; READING FROM TEXT SCREEN?
8DB2  C945                CMP    #'E'
8DB4  D003 ^8DB9          BNE    :XI045             ; NO.

8DB6  20A79F              JSR    CRSNOP             ; DISABLE CURSOR AGAIN (A = $45).

8DB9  A901       :XI045   LDA    #1                 ; SET ACCEPT LITERAL.
8DBB  8D4605              STA    AXFLAG
8DBE  20B88B              JSR    SAVIT              ; SAVE NAME IF STRING TARGET.
8DC1  4CC785              JMP    XAC024             ; GO TO ACCEPT CODE TO FINISH PROCESSING.
```

```
8DC4                        PROC
                    ;
                    ; XOUT -- WRITE COMMAND PROCESSOR
                    ;
8DC4  A908          XOUT     LDA     #OWRIT           ; WRITE DIRECTION.
8DC6  20FD97                 JSR     SCNDEV           ; CONVERT I/O SPEC TO DEVICE INDEX.
8DC9  D02B ^8DF6             BNE     :X0090           ; ERROR.

8DCB  86AB                   STX     XTEMP            ; SAVE IOCB INDEX.
8DCD  B180                   LDA     (INLN),Y
8DCF  20F99E                 JSR     CHKTRM           ; TERMINATOR FOLLOWING DEVICE SPEC?
8DD2  F001 ^8DD5             BEQ     :X0005           ; YES -- DON'T ADVANCE INDEX.

8DD4  C8                     INY                      ; NO -- SKIP OVER SINGLE SEPARATOR.

8DD5  20A7A0        :X0005   JSR     TEXP             ; REST OF STATEMENT IS A TEXT EXPRESSION.

                    ; *S*    LDA     EXEC             ; EXECUTE MODE?
8DD8  F01C ^8DF6             BEQ     :X0090           ; NO.

8DDA  84AC                   STY     XTEMP+1          ; SAVE LINE INDEX.
8DDC  A6AB                   LDX     XTEMP            ; GET IOCB INDEX.
8DDE  A48E                   LDY     TELN+2           ; START OF TEXT EXPRESSION EVALUATION.

8DE0  C48F          :X0010   CPY     TELN+3           ; DONE?
8DE2  F009 ^8DED             BEQ     :X0020           ; YES.

8DE4  B9008C                 LDA     TEXBUF,Y         ; NO -- PUT CHAR TO DEVICE.
8DE7  205897                 JSR     DOUT
8DEA  C8                     INY
8DEB  D0F3 ^8DE0             BNE     :X0010           ; (BRA).

8DED  A99B          :X0020   LDA     #EOL             ; TERMINATE RECORD.
8DEF  205897                 JSR     DOUT

8DF2  A4AC                   LDY     XTEMP+1
8DF4  A900                   LDA     #0               ; SET CC FOR NORMAL EXIT.

8DF6  60            :X0090   RTS                      ; RETURN WITH CC SET.
```

```
8DF7                     PROC
                   ;
                   ; XDONE -- CLOSE COMMAND PROCESSOR
                   ;
8DF7  A900         XDONE  LDA    #0              ; INVALID OPEN CODE MEANS CLOSE.
8DF9  20FD97              JSR    SCNDEV          ; CONVERT DEVICE SPEC TO IOCB INDEX.
8DFC  D009 ^8E07          BNE    :XD090          ; ERROR.

8DFE  A592                LDA    EXEC            ; EXECUTE MODE?
8E00  F005 ^8E07          BEQ    :XD090          ; NO.

8E02  203F97              JSR    DCLOSE          ; YES -- CLOSE IOCB & DEVICE.

8E05  A900                LDA    #0              ; SET CC FOR NORMAL EXIT.

8E07  60           :XD090 RTS                    ; RETURN WITH CC SET.
```

```
8E08                         PROC
                       ;
                       ; XSSAV -- SAVE SCREEN COMMAND PROCESSOR.
                       ;
8E08  20C297           XSSAV  JSR     SFNAME            ; EXTRACT DEVICE/FILENAME.
8E0B  D050 ^8E5D              BNE     :XS090            ; ERROR.

8E0D  20079F                  JSR     SKPSEF            ; SKIP SEPARATOR(S).

8E10  A592                    LDA     EXEC              ; EXECUTE MODE?
8E12  F049 ^8E5D              BEQ     :XS090            ; NO.

8E14  A230                    LDX     #IOCB3            ; YES -- OPEN DEVICE FOR OUTPUT.
8E16  A908                    LDA     #OWRIT
8E18  20F496                  JSR     DOPEN

8E1B  A90B                    LDA     #PUTC             ; SETUP IOCB FOR PUT CHARACTER.
8E1D  9D4203                  STA     ICCOM,X

8E20  A230                    LDX     #IOCB3
8E22  AD1405                  LDA     GRFLAG            ; GRAPHICS SCREEN FLAG.
8E25  205897                  JSR     DOUT
8E28  AD3705                  LDA     CSMODE            ; SAVE SCREEN MODE.
8E2B  205897                  JSR     DOUT
8E2E  AD5205                  LDA     SPLTSC            ; FULL/SPLIT FLAG.
8E31  205897                  JSR     DOUT
8E34  AD5105                  LDA     LETTRSZ           ; LETTER SIZE.
8E37  205897                  JSR     DOUT

8E3A  A559                    LDA     SAVMSC+1          ; SETUP POINTER TO BOTTOM OF SCREEN.
8E3C  85F7                    STA     ADRESS+1
8E3E  A900                    LDA     #0
8E40  85F6                    STA     ADRESS
8E42  844B                    STY     XTEMP
8E44  A458                    LDY     SAVMSC

8E46  B1F6             :XS010 LDA     (ADRESS),Y        ; GET DATA BYTE.
8E48  205897                  JSR     DOUT              ; OUTPUT IT.
8E4B  C8                      INY
8E4C  D0F8 ^8E46              BNE     :XS010

8E4E  E6F7                    INC     ADRESS+1
8E50  A5F7                    LDA     ADRESS+1
8E52  C56A                    CMP     RAMTOP            ; DONE?
8E54  D0F0 ^8E46              BNE     :XS010            ; NO.

8E56  A44B                    LDY     XTEMP
8E58  203F97                  JSR     DCLOSE

8E5B  A900                    LDA     #0                ; SET CC FOR NORMAL EXIT.

8E5D  60               :XS090 RTS                       ; RETURN WITH CC SET.

8E5E                          PROC
                       ;
                       ; XSLOAD -- LOAD SCREEN COMMAND PROCESSOR.
                       ;
```

```
8E5E  20C597      XSLOD   JSR    SFNAME         ; EXTRACT DEVICE/FILENAME.
8E61  D05F ^8EC2          BNE    :XS090         ; ERROR.

8E63  A592                LDA    EXEC           ; EXECUTE MODE?
8E65  F05B ^8EC2          BEQ    :XS090         ; NO.

8E67  A230                LDX    #IOCB3         ; YES -- OPEN DEVICE FOR INPUT.
8E69  A904                LDA    #OREAD
8E6B  20F496              JSR    DOPEN

8E6E  84AE                STY    XTEMP
8E70  A907                LDA    #GETC          ; SETUP IOCB FOR GET CHARACTER.
8E72  9D4203              STA    ICCOM,X
8E75  A230                LDX    #IOCB3
8E77  205397              JSR    DIN            ; GET GRAPHICS FLAG.
8E7A  8D1405              STA    GRFLAG
8E7D  205397              JSR    DIN            ; GET SCREEN MODE.
8E80  8D3705              STA    GSMODE
8E83  205397              JSR    DIN            ; GET FULL/SPLIT FLAG.
8E86  8D5205              STA    SPLTSC
8E89  205397              JSR    DIN            ; GET LETTER SIZE.
8E8C  8D5105              STA    LETTRSZ
8E8F  20B896              JSR    TSTMOD         ; SEE IF TEXT/SMALL LETTERS.
8E92  C901                CMP    #TXSL
8E94  D006 ^8E9C          BNE    :XS005         ; NO.

8E96  20F494              JSR    TXOPEN         ; YES.
8E99  4C9F8E              JMP    :XS007

8E9C  201095      :XS005  JSR    GSOPEN         ; OPEN SCREEN.

8E9F  A559        :XS007  LDA    SAVMSC+1       ; SETUP POINTER TO BOTTOM OF SCREEN.
8EA1  85F7                STA    ADRESS+1
8EA3  A900                LDA    #0
8EA5  85F6                STA    ADRESS
8EA7  A230                LDX    #IOCB3
8EA9  A458                LDY    SAVMSC

8EAB  205397      :XS010  JSR    DIN
8EAE  91F6                STA    (ADRESS),Y
8EB0  C8                  INY
8EB1  D0F8 ^8EAB          BNE    :XS010

8EB3  E6F7                INC    ADRESS+1
8EB5  A5F7                LDA    ADRESS+1
8EB7  C56A                CMP    RAMTOP         ; DONE?
8EB9  D0F0 ^8EAB          BNE    :XS010         ; NO.

8EBB  A4AE                LDY    XTEMP
8EBD  203F97              JSR    DCLOSE

8EC0  A900                LDA    #0

8EC2  60          :XS090  RTS
```

```
8EC3                           PROC

                        ;
                        ; XDIR -- DISK DIRECTORY COMMAND PROCESSOR
                        ;

8EC3  20049F      XDIR    JSR     EXP               ; GET DRIVE NUMBER.

8EC6  A592                LDA     EXEC              ; EXECUTE MODE?
8EC8  F034 ^8EFE          BEQ     :XD090            ; NO -- SYNTAX SCAN ONLY.

8ECA  A207                LDX     #:DTLNG           ; MOVE OPEN TEMPLATE ...

8ECC  BDFE8E      :XD005  LDA     :DIRTB-1,X        ; ... TO OPEN BUFFER.
8ECF  9D1F05              STA     OPNBUF-1,X
8ED2  CA                  DEX
8ED3  D0F7 ^8ECC          BNE     :XD005

8ED5  A593                LDA     EXPSTK            ; INSERT DRIVE #.
8ED7  0930                ORA     #'0'
8ED9  8D2105              STA     OPNBUF+1

8EDC  A230                LDX     #IOCB3
8EDE  A906                LDA     #DREAD+2          ; OPEN FOR DIRECTORY READ.
8EE0  20F496              JSR     DOPEN

8EE3  A230        :XD010  LDX     #IOCB3            ; GET A BYTE.
8EE5  205397              JSR     DIN

8EE8  A6E4                LDX     IOSTAT            ; CHECK FOR END-OF-FILE.
8EEA  E088                CPX     #$88
8EEC  F006 ^8EF4          BEQ     :XD020            ; EOF -- ALL DONE.

8EEE  208294              JSR     CHOT              ; WRITE TO SCREEN.
8EF1  4CE38E              JMP     :XD010

8EF4  A230        :XD020  LDX     #IOCB3            ; CLOSE THE FILE.
8EF6  203F97              JSR     DCLOSE
8EF9  20989F              JSR     NEWLIN

8EFC  A900                LDA     #0                ; SET CC FOR EXIT.

8EFE  60          :XD090  RTS

8EFF  44203A2A2E  :DIRTB  DB      'D :*.*',EOL      ; DIRECTORY OPEN TEMPLATE.
      = 0007      :DTLNG  = *-:DIRTB


8F06                           PROC
                        ;
                        ; XCOMM -- COMMAND TABLE LISTER
                        ;

8F06  F014 ^8F1C  XCOMM   BEQ     :XC090            ; SYNTAX SCAN ONLY.

8F08  AD0005              LDA     USRTAB            ; FIRST LIST USER SUPPLIED TABLE.
8F0B  AE0105              LDX     USRTAB+1
```

```
8F0E  F003 ^8F13           BEQ     :XC010          ; NO TABLE.

8F10  20108F               JSR     PRINTC

8F13  A9D1        :XC010   LDA     # LOW CTAB      ; NOW LIST BUILT-IN TABLE.
8F15  A27D                 LDX     # HIGH CTAB
8F17  20108F               JSR     PRINTC

8F1A  A900                 LDA     #0

8F1C  60          :XC090   RTS                     ; RETURN WITH CC SET.

8F1D  8590        PRINTC   STA     TABADR          ; SETUP POINTER TO BEGIN OF TABLE.
8F1F  8691                 STX     TABADR+1
8F21  84AB                 STY     XTEMP           ; SAVE Y REG.

8F23  A905        :PC003   LDA     #5              ; 5 NAMES PER LINE.
8F25  85AC                 STA     XTEMP+1

8F27  A000        :PC005   LDY     #0              ; START NAME SCAN.

8F29  B190        :PC010   LDA     (TABADR),Y      ; GET CHARACTER.
8F2B  F021 ^8F4E           BEQ     :PC080          ; END OF TABLE.

8F2D  3006 ^8F35           BMI     :PC020          ; END OF NAME.

8F2F  208294               JSR     CHOT            ; OUTPUT CHARACTER.
8F32  C8                   INY
8F33  D0F4 ^8F29           BNE     :PC010          ; (BRA).

8F35  209D9F      :PC020   JSR     SPACES
8F38  C8                   INY                     ; SKIP OVER PARAMETERS.
8F39  C8                   INY
8F3A  98                   TYA                     ; ADD Y REG TO TABADR.
8F3B  18                   CLC
8F3C  6590                 ADC     TABADR
8F3E  8590                 STA     TABADR
8F40  9002 ^8F44           BCC     :PC030

8F42  E691                 INC     TABADR+1

8F44  C6AC        :PC030   DEC     XTEMP+1         ; LINE FULL (5 NAMES)?
8F46  D0DF ^8F27           BNE     :PC005

8F48  20969F               JSR     NEWLIN          ; YES.
8F4B  4C238F               JMP     :PC003

8F4E  20969F      :PC080   JSR     NEWLIN
8F51  A4AB                 LDY     XTEMP
8F53  60                   RTS
```

```
8F54                        PROC
                    ;
                    ; XWAIT -- PAUSE COMMAND PROCESSOR
                    ;
8F54  20D49F        XWAIT   JSR     EXP             ; THERE MUST BE AN EXPRESSION FOLLOWING.

8F57  A592                  LDA     EXEC            ; EXECUTE MODE?
8F59  F01B ^8F76            BEQ     :XW090          ; NO -- ALL DONE.

8F5B  A213                  LDX     #EXPSTK-DTAB    ; YES -- WORK WITH COUNT.
8F5D  84AE                  STY     XTEMP           ; SAVE LINE INDEX.

8F5F  A4AE          :XW010  LDY     XTEMP           ; RESTORE INDEX.
8F61  A593                  LDA     EXPSTK          ; ALL DONE?
8F63  0594                  ORA     EXPSTK+1
8F65  F00F ^8F76            BEQ     :XW090          ; YES.

8F67  A414                  LDY     RTCLOK+2        ; NO -- WAIT FOR ...

8F69  207E9F        :XW020  JSR     ABRTCK          ; ... OPERATOR ABORT ...
8F6C  C414                  CPY     RTCLOK+2
8F6E  F0F9 ^8F69            BEQ     :XW020          ; ... OR CLOCK TO CHANGE.

8F70  20129D                JSR     DDCHI           ; DECREMENT COUNT.
8F73  4C5F8F                JMP     :XW010

8F76  60            :XW090  RTS

8F77                        PROC
                    ;
                    ; XSPEED -- SPEED CONTROL COMMAND PROCESSOR
                    ;
8F77  20D49F        XSPEED  JSR     EXP             ; THERE MUST BE AN EXPRESSION.

8F7A  A592                  LDA     EXEC            ; EXECUTE MODE?
8F7C  F007 ^8F85            BEQ     :XS090          ; NO.

8F7E  A593                  LDA     EXPSTK          ; YES -- SET SPEED.
8F80  8D5005                STA     SPEED

8F83  A900                  LDA     #0              ; SET CC FOR EXIT.

8F85  60            :XS090  RTS                     ; RETURN WITH CC SET.
```

```
8F86                          PROC
                    ;
                    ; XCASS -- CASSETTE ON/OFF COMMAND PROCESSOR
                    ;
8F86  A20A          XCASS     LDX     #ONOFFX          ; CHECK FOR 'ON' OR 'OFF'.
8F88  20A87C                  JSR     SBCMAT
8F8B  D00C ^8F99              BNE     :XC090           ; NOT FOUND -- ERROR.

8F8D  A592                    LDA     EXEC             ; EXECUTE MODE?
8F8F  F008 ^8F99              BEQ     :XC090           ; NO.

8F91  BD9A8F                  LDA     CASCTL,X         ; 0/1 -> CASSETTE CONTROL.
8F94  8D02D3                  STA     PACTL

8F97  A900                    LDA     #0               ; SET CC FOR NORMAL EXIT.

8F99  60            :XC090    RTS                      ; RETURN WITH CC SET.

                    ; CASSETTE CONTROL
                    ; REQUIRES KOFF = 0, KON = 1.

8F9A  3C            CASCTL    DB      CASSOF
8F9B  34                      DB      CASSON


8F9C                          PROC
                    ;
                    ; XCSYNC -- CASSETTE SYNC COMMAND PROCESSOR
                    ;
8F9C  F01B ^8FB9    XCSYNC    BEQ     :XC090           ; SYNTAX SCAN.

8F9E  AD02D3                  LDA     PACTL            ; CHECK CASSETTE MOTOR.
8FA1  2908                    AND     #$08
8FA3  D012 ^8FB7              BNE     :XC080           ; MOTOR OFF.

8FA5  A910                    LDA     #$10             ; ON -- WAIT FOR MARK TO SPACE TRANSITION.

8FA7  207E9F        :XC010    JSR     ABRTCK           ; WAIT FOR BREAK ...
8FAA  2C0FD2                  BIT     SKSTAT
8FAD  F0F8 ^8FA7              BEQ     :XC010           ; ... OR MARK.

8FAF  207E9F        :XC020    JSR     ABRTCK           ; WAIT FOR BREAK ...
8FB2  2C0FD2                  BIT     SKSTAT
8FB5  D0F8 ^8FAF              BNE     :XC020           ; ... OR SPACE.

8FB7  A900          :XC080    LDA     #0               ; SET CC FOR NORMAL EXIT.

8FB9  60            :XC090    RTS                      ; RETURN WITH CC SET.


8FBA                          PROC
                    ;
                    ; XTRACE -- TRACE MODE ON/OFF COMMAND
                    ;
8FBA  A20A          XTRACE    LDX     #ONOFFX          ; CHECK FOR 'ON' OR 'OFF'.
```

```
8FBC  20A87C                JSR     SBCMAT
8FBF  D009 ^8FCA            BNE     :XT090          ; NOT FOUND -- ERROR.

8FC1  A592                  LDA     EXEC            ; EXECUTE MODE?
8FC3  F005 ^8FCA            BEQ     :XT090          ; NO.

                    ; REQUIRES KOFF = 0, KON <> 0.

8FC5  8E3505                STX     TRACE           ; SET FLAG.
8FC8  A900                  LDA     #0              ; SET CC FOR NORMAL EXIT.

8FCA  60            :XT090  RTS                     ; RETURN WITH CC SET.
```

```
                    ATARI CAMAC Assembler Ver  1.0A  Page 109
 PILOT -- H.B. STEWART                                    D1:PILOT.

 8FCB                              PROC
                          ;
                          ; XSAVE -- SAVE COMMAND PROCESSOR
                          ;
 8FCB  20D696             XSAVE    JSR    CNAME            ; EXTRACT DEVICE/FILENAME.
 8FCE  20679F                      JSR    SKPSEP           ; SKIP SEPARATOR(S).

 8FD1  A592                        LDA    EXEC             ; EXECUTE MODE?
 8FD3  F024 ^8FF9                  BEQ    :XS090           ; NO.

 8FD5  A230                        LDX    #IOCB3
 8FD7  A908                        LDA    #OWRIT           ; YES -- OPEN DEVICE FOR OUTPUT.
 8FD9  20F496                      JSR    DOPEN

 8FDC  A90B                        LDA    #PUTC            ; SETUP IOCB FOR PUT CHARACTER.
 8FDE  9D4203                      STA    ICCOM,X

 8FE1  A980                        LDA    #$80+IOCB3       ; RE-ROUTE 'CHOT' OUTPUT TO DEVICE.
 8FE3  8D3005                      STA    CDEST

 8FE6  208490                      JSR    LISTER           ; OUTPUT PROGRAM TO DEVICE.

 8FE9  A230                        LDX    #IOCB3
 8FEB  203F97                      JSR    DCLOSE           ; CLOSE DEVICE.

 8FEE  A906                        LDA    #EPUTC-IOVBAS    ; RESTORE 'CHOT' OUTPUT.
 8FF0  8D3005                      STA    CDEST

                          ; *** EXTERNAL ENTRY POINT FROM 'XLIST' ***

 8FF3  202C85             XSV050   JSR    RDYMES           ; GENERATE "READY" MESSAGE.

 8FF6  A900                        LDA    #0               ; SET CC FOR NORMAL EXIT.

 8FF8                     XAP090
 8FF8                     XME090
 8FF8                     XLD090
 8FF8  60                          RTS                     ; RETURN WITH CC SET.

 8FF9  4C8490             :XS090   JMP    LISTER           ; SYNTAX CHECK & RETURN WITH CC SET.
```

```
8FFC                          PROC
                        ;
                        ; XLOAD -- LOAD COMMAND PROCESSOR
                        ;

8FFC  201090            XLOAD   JSR     XL0100          ; COMMON CODE.
8FFF  D0F7 ^8FF8                BNE     XLD090          ; ERROR.

                        ; *** EXTERNAL ENTRY FROM 'XRUN' ***

9001  A592              XL0005  LDA     EXEC            ; EXECUTE MODE?
9003  F0F3 ^8FF8                BEQ     XLD090          ; NOT.

9005  20C087                    JSR     CLRPRG          ; CLEAR PROGRAM STORAGE AREA.
9008  A901                      LDA     #KLOAD          ; SET LOAD FLAG.

                        ; *** EXTERNAL ENTRY FROM 'XMERGE', 'XAPPND' ***

900A  8D3205            XL0010  STA     LOADFG

900D  4C0979                    JMP     MLLOAD          ; LOAD UNTIL I/O ERROR OR END OF FILE.
                                                        ; SET 'GETCOM'.

9010  20C297            XL0100  JSR     SFNAME          ; EXTRACT DEVICE/FILENAME.
9013  D0E3 ^8FF8                BNE     XLD090          ; ERROR.

9015  A592                      LDA     EXEC            ; EXECUTE MODE?
9017  F0DF ^8FF8                BEQ     XLD090          ; NO.

9019  AD3205                    LDA     LOADFG          ; ALREADY LOADING?
901C  D0DA ^8FF8                BNE     XLD090          ; YES -- ERROR.

901E  A230                      LDX     #IOCB3
9020  A904                      LDA     #OREAD          ; YES -- OPEN DEVICE FOR READING.
9022  20F496                    JSR     DOPEN

9025  A900                      LDA     #0              ; CLEAR USE STACK.
9027  8D4D05                    STA     USTKP

902A  60                        RTS



                        ;
                        ; XMERGE -- MERGE COMMAND PROCESSOR.
                        ;

902B  201090            XMERGE  JSR     XL0100          ; COMMON CODE.
902E  D0C8 ^8FF8                BNE     XME090          ; ERROR.

9030  A592                      LDA     EXEC            ; EXECUTE MODE?
9032  F0C4 ^8FF8                BEQ     XME090          ; NO.

9034  A902                      LDA     #KMERGE         ; SET LOAD FLAG.
9036  D0D2 ^900A                BNE     XL0010          ; (BRA).
```

```
                       ;
                       ; XAPPND -- APPEND COMMAND PROCESSOR.
                       ;

9038  201090           XAPPND  JSR     XLD100          ; COMMON CODE.
903B  D0BB ^8FF8               BNE     XAP090          ; ERROR.

903D  20B891                   JSR     XAU010          ; SHARE 'XAUTO' CODE FOR LINE #'S.
9040  D0B6 ^8FF8               BNE     XAP090          ; ERROR.

9042  A592                     LDA     EXEC            ; EXECUTE MODE?
9044  F0B2 ^8FF8               BEQ     XAP090          ; NO.

9046  A903                     LDA     #KAPPND         ; SET LOAD FLAG.
9048  D0C0 ^900A               BNE     XLD010          ; (BRA).
```

```
904A                        PROC
                     ;
                     ; XLETTR -- TEXT LETTER SIZE SELECTION
                     ;
904A  A20C           XLETTR  LDX     #LTTABX          ; CHECK FOR 'SMALL', 'MEDIUM', OR 'LARGE'.
904C  20AB7C                 JSR     SBCMAT
904F  D02D ^907E             BNE     :XL090           ; NOT FOUND -- ERROR.

9051  A592                   LDA     EXEC             ; EXECUTE MODE?
9053  F029 ^907E             BEQ     :XL090           ; NO.

9055  AD4505                 LDA     SGLSTP           ; SINGLE STEP?
9058  F004 ^905E             BEQ     :XL020           ; NO.

905A  A983                   LDA     #NRCERR          ; YES -- ERROR.
905C  D020 ^907E             BNE     :XL090

905E  84AB           :XL020  STY     XTEMP
9060  8E5105                 STX     LETTRSZ          ; YES -- SET NEW LETTER SIZE.
9063  8A                     TXA
9064  D006 ^906C             BNE     :XL050           ; MEDIUM OR LARGE LETTERS.

9066  20F494                 JSR     TXOPEN           ; SMALL LETTERS.
9069  4C7790                 JMP     :XL080

906C  8E3705         :XL050  STX     GSMODE           ; GRAPHICS MODE.
906F  A900                   LDA     #0
9071  8D5205                 STA     SPLTSC           ; NO SPLIT SCREEN.

9074  201095                 JSR     GSOPEN           ; OPEN SCREEN.

9077  A4AB           :XL080  LDY     XTEMP
9079  A900                   LDA     #0               ; RESET GRAPHICS MODE FLAG & SET CC.
907B  8D1405                 STA     GRFLAG

907E  60             :XL090  RTS                      ; RETURN WITH CC SET.
907F                         PROC
                     ;
                     ; XSCROLL -- SCROLL OPTION SELECTION
                     ;

907F  A210           XSCROLL LDX     #SCTABX          ; CHECK FOR 'FINE' OR 'COARSE'.
9081  20AB7C                 JSR     SBCMAT
9084  D01D ^90A3             BNE     :XS090           ; NOT FOUND -- ERROR.

9086  A592                   LDA     EXEC             ; EXECUTE MODE?
9088  F019 ^90A3             BEQ     :XS090           ; NO.

908A  20BB96                 JSR     TSTMOD           ; TEXT MODE, SMALL LETTERS?
908D  C901                   CMP     #TXSL
908F  D013 ^90A4             BNE     :XS092           ; NO.

9091  8EF205                 STX     FINEFG           ; SET SCREEN EDITOR FLAG.

9094  84AB                   STY     XTEMP
9096  202996                 JSR     COMPRS           ; COMPRESS MEMORY.
9099  208E96                 JSR     EOPEN            ; RE-OPEN E: ON IOCB 0.
```

```
909C  205C96             JSR    EXPAND          ; EXPAND MEMORY.
909F  A4AB               LDY    XTEMP

90A1  A900               LDA    #0              ; SET CC FOR NORMAL EXIT.

90A3  60         :XS090  RTS                    ; RETURN WITH CC SET.

90A4  A983       :XS092  LDA    #NRCERR
90A6  60                 RTS
```

```
90A7                          PROC
                    ;
                    ; XLIST -- LIST COMMAND PROCESSOR
                    ;

                    ; *** EXTERNAL ENTRY POINT FROM 'XSAVE' ***

90A7  20B490        XLIST   JSR     LISTER          ; DO THE LIST PROCESS.
90AA  D007 ^90B3            BNE     :XL009          ; ERROR.

90AC  A592                  LDA     EXEC            ; EXECUTE MODE?
90AE  F003 ^90B3            BEQ     :XL009          ; NO.
90B0  4CF38F                JMP     XSV050          ; YES -- SIGN OFF & RETURN.

90B3  60            :XL009  RTS                     ; RETURN WITH CC SET.

90B4  A90D          LISTER  LDA     #LOW LSTNMS     ; ADDRESS OF DEFAULTS.
90B6  85B6                  STA     POINT
90B8  A991                  LDA     #HIGH LSTNMS
90BA  85B7                  STA     POINT+1
90BC  206D93                JSR     MNYNMS          ; GET PARAMETERS.
90BF  D045 ^9106            BNE     :XL900          ; SYNTAX ERROR.

90C1  E003                  CPX     #3
90C3  B041 ^9106            BCS     :XL900          ; TOO MANY NUMBERS.

90C5  E001                  CPX     #1              ; HOW MANY ARGS?
90C7  D00C ^90D5            BNE     :XL010          ; 0 OR 2.

90C9  AD5305                LDA     NMSBF           ; 1 -- LAST LINE = FIRST.
90CC  8D5505                STA     NMSBF+2
90CF  AD5405                LDA     NMSBF+1
90D2  8D5605                STA     NMSBF+3

90D5                :XL010

                    ; *S*   STY     XTEMP           ; SAVE Y.
90D5  A25E                  LDX     #LS-DTAB        ; 'LS'= FIRST.
90D7  A000                  LDY     #0
90D9  207794                JSR     NMOVI

90DC  A260                  LDX     #LEND-DTAB      ; 'LEND' = SECOND.
90DE  A002                  LDY     #2
90E0  207794                JSR     NMOVI

90E3  20C893                JSR     BRACKT          ; BRACKET RANGE.
90E6  D01E ^9106            BNE     :XL900          ; FIRST > LAST.

90E8  A592                  LDA     EXEC            ; EXECUTE MODE?
90EA  F01C ^9108            BEQ     :XL990          ; NO.

90EC  A213          :XL100  LDX     #BLOW-DTAB      ; ADDRESS OF NEXT LINE
90EE  A015                  LDY     #BHIGH-DTAB     ; ADDRESS PAST END.
90F0  20159C                JSR     DCMPI
90F3  B00D ^9102            BCS     :XL200          ; DONE.

90F5  A013                  LDY     #BLOW-DTAB
```

```
90F7  20229F                JSR     PSF              ; PRINT STORAGE FORM LINE.

90FA  A213                  LDX     #BLCK-DTAB       ; ADVANCE TO NEXT LINE.
90FC  20AA9A                JSR     SNXTI
90FF  4CEC90                JMP     :XL100

9102  A900        :XL200    LDA     #0               ; SET CC FOR NORMAL EXIT.
9104  F002 ^9108            BEQ     :XL990           ; (BRA).

9106  A902        :XL900    LDA     #IMPERR          ; IMPROPER PARAMETER ERROR.

9108  08          :XL990    PHP                      ; SAVE CC.
9109  A4AB                  LDY     XTEMP            ; RESTORE Y.
910B  28                    PLP                      ; RESTORE CC.
910C  60                    RTS                      ; RETURN WITH CC SET.

                  ; DEFAULTS FOR 'LIST'.

910D  0000        LSTNMS    DW      0
910F  0F27                  DW      MAXLN
9111  FFFF                  DW      EONMLS
```

```
9113                          PROC
                       ;
                       ; XAUTO -- AUTO-INPUT COMMAND PROCESSOR
                       ;

9113  201A91           XAUTO   JSR     XAU010           ; COMMON CODE.
9116  D04A ^9162              BNE     :XA900           ; ERROR.
9118  F030 ^914A               BEQ     :XA200

                       ; *** EXTERNAL ENTRY FOR 'APPEND' ***

911A  A969             XAU010  LDA     #LOW AUTNMS      ; ADDRESS OF DEFAULTS.
911C  85E6                     STA     POINT
911E  A991                     LDA     #HIGH AUTNMS
9120  85E7                     STA     POINT+1
9122  206D93                   JSR     MNYNMS           ; GET PARAMETERS.
9125  D022 ^9149               BNE     :XA190           ; SYNTAX ERROR.

                       ; *S*   STY     XTEMP            ; SAVE Y.
9127  8A                       TXA                      ; SET 'Z' FLAG.
9128  D00D ^9137               BNE     :XA100           ; FIRST LINE ENTERED.

                       ; DEFAULTS: NEXT LINE = LAST PROGRAM LINE + 10

912A  201294                   JSR     GTLSLN           ; 'LINENO' = LAST PROGRAM LINE + 10.
912D  A27A                     LDX     #ALINE-DTAB
912F  A05C                     LDY     #LINENO-DTAB
9131  20459A                   JSR     DMOVI
9134  4C3E91                   JMP     :XA110

                       ; USE ENTERED VALUES.

9137  A27A             :XA100  LDX     #ALINE-DTAB      ; 'ALINE' = FIRST.
9139  A000                     LDY     #0
913B  207794                   JSR     NMOVI

913E  A27C             :XA110  LDX     #AINC-DTAB       ; 'AINC' = SECOND.
9140  A002                     LDY     #2
9142  207794                   JSR     NMOVI
9145  A4AE                     LDY     XTEMP            ; RESTORE Y.
9147  A900                     LDA     #0               ; SET CC FOR EXIT.

9149  60               :XA190  RTS

914A  A592             :XA200  LDA     EXEC             ; EXECUTE MODE?
914C  F016 ^9164               BEQ     :XA990           ; NO.

914E  8D3605                   STA     AUTOIN           ; YES -- SET AUTO-INPUT MODE.
9151  A586                     LDA     ACOLR2           ; SET SCREEN BACKGROUND COLOR.
9153  8DC602                   STA     COLOR0+2
9156  A587                     LDA     ACOLR1           ; SET SCREEN LETTER COLOR.
9158  8DC502                   STA     COLOR0+1

915B  A900                     LDA     #0               ; SET CC FOR NORMAL EXIT.
915D  8DCA05                   STA     INDENT           ; INITIALIZE 'AUTO INDENT'.

9160  F002 ^9164               BEQ     :XA990           ; (BRA).
```

```
9162  A902         :XA900  LDA     #INFERR

9164  08           :XA990  PHP                           ; SAVE CC.
9165  A4AB                 LDY     XTEMP                 ; RESTORE Y.
9167  28                   PLP                           ; RESTORE CC.
9168  60                   RTS

                   ; DEFAULTS FOR 'AUTO', 'APPEND'

9169  0000         AUTNMS  DW      0                     ; (DON'T CARE).
916B  0A00                 DW      10
916D  FFFF                 DW      EONMLS
```

```
916F                          PROC
                         ;
                         ; XDELET -- DELETE COMMAND PROCESSOR
                         ;
916F  A900               XDELET  LDA    #LOW LSTNMS       ; SHARE DEFAULTS.
9171  85B6                       STA    POINT
9173  A991                       LDA    #HIGH LSTNMS
9175  85B7                       STA    POINT+1
9177  206D93                     JSR    MNYNMS            ; GET PARAMETERS.
917A  D07C ^91F8                 BNE    :XD900            ; SYNTAX ERROR.

917C  E001                       CPX    #1                ; 0, 1, OR 2 PARAMETERS.
917E  9078 ^91F8                 BCC    :XD900            ; 0 = ERROR.
9180  D00C ^918E                 BNE    :XD010            ; 2.

9182  AD5305                     LDA    NMSBF             ; 1 -- LAST LINE = FIRST.
9185  8D5505                     STA    NMSBF+2
9188  AD5405                     LDA    NMSBF+1
918B  8D5605                     STA    NMSBF+3

918E                     :XD010
                         ; *S*   STY    XTEMP             ; SAVE Y.
918E  A25E                       LDX    #LS-DTAB          ; 'LS' = FIRST.
9190  A000                       LDY    #0
9192  207794                     JSR    NMOVI

9195  A260                       LDX    #LEND-DTAB        ; 'LEND' = SECOND.
9197  A002                       LDY    #2
9199  207794                     JSR    NMOVI

919C  20C893                     JSR    BRACKT            ; BRACKET RANGE.
919F  D057 ^91F8                 BNE    :XD900            ; FIRST > LAST.

91A1  A592                       LDA    EXEC              ; EXECUTE MODE?
91A3  F055 ^91FA                 BEQ    :XD990            ; NO.

91A5  A597                       LDA    BNUM              ; ANY LINES TO DELETE?
91A7  0598                       ORA    BNUM+1
91A9  F049 ^91F4                 BEQ    :XD600            ; NO.

                         ; WARN USER.

91AB  A99F                       LDA    #DELMES           ; 'YOU ARE ABOUT TO DELETE '.
91AD  20FFB4                     JSR    MESSOT

91B0  A217                       LDX    #BNUM-DTAB        ; # OF LINES.
91B2  20149E                     JSR    DECASC

91B5  A9A0                       LDA    #DL2MES           ; 'LINES(S).<CR> ARE YOU SURE?'
91B7  20FFB4                     JSR    MESSOT

91BA  A20C                       LDX    #TELN-DTAB        ; USE 'TEXBUF'.
91BC  20B194                     JSR    GETLIN

91BF  A58F                       LDA    TELN+3            ; EMPTY?
91C1  F02C ^91EF                 BEQ    :XD500            ; YES -- DO NOT CHANGE.
```

```
 91C3  AD008C                 LDA     TEXBUF          ; FIRST CHARACTER.
 91C6  0920                   ORA     #LC             ; FORCE LOWER CASE.
 91C8  C979                   CMP     #'Y'+$20        ; Y?
 91CA  D023 ^91EF             BNE     :XD500          ; NO -- DO NOT CHANGE.

                      ; USER AGREES.

 91CC  8D4305                 STA     NOCCNT          ; NO CONTINUE AFTER DELETIONS.
 91CF  A215                   LDX     #BHIGH-DTAB     ; SIZE OF BRACKETED RANGE.
 91D1  A013                   LDY     #BLOW-DTAB
 91D3  20429C                 JSR     DSUBI

 91D6  A252                   LDX     #MEMA-DTAB      ; GET READY TO DELETE.
                      ; *S*     LDY     #BLOW-DTAB
 91D8  20459A                 JSR     DMOVI

 91DB  A000                   LDY     #0              ; SET BLOCK SIZE TO DELETE.
 91DD  A595                   LDA     BHIGH
 91DF  9193                   STA     (BLOW),Y
 91E1  C8                     INY
 91E2  A596                   LDA     BHIGH+1
 91E4  9193                   STA     (BLOW),Y

 91E6  203E9B                 JSR     MDEALL          ; DELETE BLOCK.

 91E9  202C85                 JSR     RDYMES
                      ;       LDA     #0              ; SET CC FOR NORMAL EXIT.
                      ;       BEQ     :XD990          ; (BRA).
 91EC  4CF491                 JMP     :XD600

                      ; USER DOES NOT AGREE.

 91EF  A99E           :XD500  LDA     #NCHGMS         ; 'PROGRAM UNCHANGED'.
 91F1  20FF84                 JSR     MESSOT

 91F4  A900           :XD600  LDA     #0              ; SET CC FOR NORMAL EXIT.
 91F6  F002 ^91FA             BEQ     :XD990

                      ; SYNTAX ERROR.

 91F8  A902           :XD900  LDA     #IMPERR         ; IMPROPER PARAMETER ERROR.

 91FA  08             :XD990  PHP                     ; SAVE CC.
 91FB  A4AB                   LDY     XTEMP           ; RESTORE Y.
 91FD  28                     PLP                     ; RESTORE CC.
 91FE  60                     RTS
```

```
91FF                        PROC
                    ;
                    ; XREN -- RENUMBER COMMAND PROCESSOR
                    ;
                    ;       STEP 1: BRACKET THE RANGE OF LINES TO RENUMBER.
                    ;            2: COMPUTE THE NEW RANGE THEY WILL BECOME.
                    ;            3: FIND STARTING AND ENDING ADDRESSES OF THE NEW LINES.
                    ;            4: THERE ARE TWO VALID CASES FOR THE NEW LINES:
                    ;                 A. THEY ALL FIT BETWEEN TWO EXISTING LINES.
                    ;                 B. THEY ALL FIT WITHIN THE RENUMBERED RANGE.
                    ;            5: RENUMBER THE LINES IN PLACE.
                    ;            6: MOVE THEM BETWEEN TWO EXISTING LINES (IF 4A.).
                    ;
91FF  A963          XREN    LDA     #LOW RENNMS     ; ADDRESS OF DEFAULTS.
9201  85B6                  STA     POINT
9203  A993                  LDA     #HIGH RENNMS
9205  85B7                  STA     POINT+1
9207  206D93                JSR     MNYNMS          ; GET PARAMETERS.
920A  F003 ^920F            BEQ     :XR010          ; OK.
920C  4C5E93        :XR005  JMP     :XR900          ; SYNTAX ERROR.

920F                :XR010
                    ; *S*   STY     XTEMP           ; SAVE Y.

920F  A27A                  LDX     #ALINE-DTAB     ; 'ALINE' = FIRST.
9211  A000                  LDY     #0
9213  207794                JSR     NMOVI

9216  A27C                  LDX     #AINC-DTAB      ; 'AINC' = SECOND.
9218  A002                  LDY     #2
921A  207794                JSR     NMOVI

921D  A25E                  LDX     #LS-DTAB        ; 'LS' = THIRD.
921F  A004                  LDY     #4
9221  207794                JSR     NMOVI

9224  A260                  LDX     #LEND-DTAB      ; 'LEND' = FOURTH.
9226  A006                  LDY     #6
9228  207794                JSR     NMOVI

922B  20C893                JSR     BRACKT          ; BRACKET RANGE.
922E  D0DC ^920C            BNE     :XR005          ; 'LS' > 'LEND'.

9230  A592                  LDA     EXEC            ; EXECUTE MODE?
9232  D003 ^9237            BNE     :XR015          ; YES.
9234  4C5E93                JMP     :XR990          ; NO.

9237  A597          :XR015  LDA     BNUM            ; 0 LINES?
9239  0598                  ORA     BNUM+1
923B  D003 ^9240            BNE     :XR020          ; NO.
923D  4C2E93                JMP     :XR500          ; YES.

9240  A219          :XR020  LDX     #RTMP-DTAB      ; 'RTMP' = # OF LINES.
9242  A017                  LDY     #BNUM-DTAB
9244  20459A                JSR     DMOVI
9247  20129D                JSR     DECWI           ; -1.
```

```
924A  A01C                    LDY     #XINC-DTAB       ; * INCREMENT.
924C  20549C                  JSR     DMULI
924F  A07A                    LDY     #ALINE-DTAB      ; + FIRST NEW LINE.
9251  20329C                  JSR     DADDI            ; = LAST NEW LINE.

9254  206894                  JSR     CHKLN            ; IS LINE IN RANGE?
9257  9003 ^925C              BCC     :XR030           ; YES.
9259  4C3593                  JMP     :XR600           ; NO -- OUT OF RANGE.

                      ; FIND STARTING AND ENDING ADDRESSES OF THE NEW RANGE.

925C                  :XR030
                      ; *S*   LDX     #RTMP-DTAB       ; 'RTMP' = LAST NEW LINE.
925C  204294                  JSR     RENFND
925F  85AC                    STA     XTEMP+1          ; SAVE 'VALID' STATUS.

9261  A21B                    LDX     #R2TMP-DTAB      ; 'R2TMP' = ADDRESS OF END.
9263  A04E                    LDY     #PP-DTAB
9265  20459A                  JSR     DMOVI

9268  A27A                    LDX     #ALINE-DTAB
926A  204294                  JSR     RENFND           ; 'PP' = ADDRESS OF START.
926D  05AC                    ORA     XTEMP+1          ; IF EITHER IS INVALID, "OVERLAP" ERROR.
926F  F003 ^9274              BEQ     :XR040           ; OK.
9271  4C4293          :XR035  JMP     :XR700           ; OVERLAP.

                      ;
                      ; OVERLAPPING RANGES UNLESS:
                      ;         'START' OF NEW = 'END' OF NEW *OR*
                      ;         'START' OF OLD <= 'START' OF NEW *AND*
                      ;         'END' OF NEW <= 'END' OF OLD
                      ;
9274  A24E            :XR040  LDX     #PP-DTAB         ; 'START' OF NEW.
9276  A01B                    LDY     #R2TMP-DTAB      ; 'END' OF NEW.
9278  20159C                  JSR     DCMPI
927B  F010 ^928D              BEQ     :XR100           ; NOT OVERLAPPING.

                      ; *S*   LDX     #PP-DTAB         ; 'START' OF NEW.
927D  A013                    LDY     #BLOW-DTAB       ; 'START' OF OLD.
927F  20159C                  JSR     DCMPI
9282  90ED ^9271              BCC     :XR035           ; OVERLAPPING.

9284  A215                    LDX     #BHIGH-DTAB      ; 'END' OF NEW.
9286  A01B                    LDY     #R2TMP-DTAB      ; 'END' OF OLD.
9288  20159C                  JSR     DCMPI
928B  90E4 ^9271              BCC     :XR035           ; OVERLAPPING.

                      ; RENUMBER IS VALID

928D  A219            :XR100  LDX     #RTMP-DTAB
928F  A04E                    LDY     #PP-DTAB
9291  20459A                  JSR     DMOVI            ; 'RTMP' = ADDRESS OF START.

                      ; RENUMBER EACH LINE IN PLACE.

9294  8D4305          :XR110  STA     NOCONT           ; NO CONTINUE AFTER RENUMBER.
9297  A24E                    LDX     #PP-DTAB
```

```
9299  A013                LDY     #BLOW-DTAB
929B  20459A              JSR     DMOVI

                  ; 'PP'     = ADDRESS OF NEXT LINE TO RENUMBER.
                  ; 'ALINE'  = NEW LINE NUMBER.
                  ; 'AINC'   = INCREMENT.
                  ; 'BNUM'   = # OF LINES LEFT TO RENUMBER.

929E  A004        :XR200  LDY     #4
92A0  A5FA                LDA     ALINE
92A2  91CE                STA     (PP),Y          ; NEW LSB (INVERTED).

92A4  88                  DEY
92A5  A5FB                LDA     ALINE+1
92A7  91CE                STA     (PP),Y          ; NEW MSB (INVERTED).

92A9  A27A                LDX     #ALINE-DTAB
92AB  A07C                LDY     #AINC-DTAB
92AD  20329C              JSR     CADDI           ; INCREMENT 'ALINE'.

92B0  A24E                LDX     #PP-DTAB
92B2  20AA9A              JSR     SNXTI           ; ADDRESS OF NEXT LINE.
92B5  A217                LDX     #BNUM-DTAB
92B7  20129D              JSR     DDCRI           ; ONE LESS LINE.

92BA  A597                LDA     BNUM            ; ANY LINES LEFT?
92BC  0598                ORA     BNUM+1
92BE  D0DE ^929E          BNE     :XR200          ; YES.

                  ; THE LINES HAVE BEEN RENUMBERED IN PLACE.

                  ; THERE ARE FOUR CASES:
                  ;     1. 'START' ADDRESS < 'END' ADDRESS -> ALREADY IN ORDER.
                  ;     2. ONE LINE MOVE, ALREADY IN ORDER (NEW = OLD).
                  ;     3. MOVE THE BLOCK TO LOWER MEMORY (NEW #'S < OLD).
                  ;     4. MOVE THE BLOCK TO HIGHER MEMORY (NEW #'S > OLD).

92C0  A219                LDX     #RTMP-DTAB      ; 'START' ADDRESS.
92C2  A01E                LDY     #R2TMP-DTAB     ; 'END' ADDRESS.
92C4  20159C              JSR     DCMPI
92C7  D065 ^932E          BNE     :XR500          ; ALREADY IN ORDER.

                  ; MOVE ONE STATEMENT AT A TIME (TO AVOID 'NOT ENOUGH MEMORY'), USING 'TEXBUF'.

                  ; 'BLOW'  = ADDRESS OF NEXT STATEMENT TO MOVE.
                  ; 'BHIGH' = ADDRESS PAST END.

                  ; *S*     LDX     #RTMP-DTAB
92C9  A013                LDY     #BLOW-DTAB
92CB  20159C              JSR     DCMPI           ; IS 'NEW' < 'OLD'?
92CE  F05E ^932E          BEQ     :XR500          ; NEW = OLD.
92D0  A900                LDA     #0              ; SET 'NEW' < 'OLD'.
92D2  9002 ^92D6          BCC     :XR210          ; YES.

92D4  A901                LDA     #1              ; NO -- SET 'NEW' > 'OLD'.
92D6  859E        :XR210  STA     R2TMP
```

```
9208  A000         :XR300  LDY     #0              ; GET LENGTH OF STATEMENT.
92DA  B193                 LDA     (BLOW),Y
92DC  A8                   TAY

92DD  B193         :XR310  LDA     (BLOW),Y        ; MOVE NEXT BYTE TO 'TEXBUF'.
92DF  99008C               STA     TEXBUF,Y        ; (EXTRA BYTE IS "DON'T CARE").
92E2  88                   DEY
92E3  10F8 ^92DD           BPL     :XR310

92E5  A252                 LDX     #MEMA-DTAB
92E7  A013                 LDY     #BLOW-DTAB
92E9  20459A               JSR     DMOVI           ; 'MEMA' = ADDRESS IN STORAGE.

92EC  203E9B               JSR     MDEALL          ; DELETE IT.

92EF  A598                 LDA     R2TMP           ; 'NEW' > 'OLD'?
92F1  D00A ^92FD           BNE     :XR320          ; YES.

                   ; 'NEW' < 'OLD'.

92F3  A213                 LDX     #BLOW-DTAB
92F5  A054                 LDY     #MEMB-DTAB
92F7  20329C               JSR     DADDI           ; ADJUST 'BLOW' FOR NEXT LINE.
92FA  4C0993               JMP     :XR330

                   ; 'NEW' > 'OLD'.

92FD  A219         :XR320  LDX     #RTMP-DTAB
92FF  A054                 LDY     #MEMB-DTAB
9301  20429C               JSR     DSUBI           ; MOVE 'RTMP' FOR INSERTION.

9304  A215                 LDX     #BHIGH-DTAB
                   ; *S*   LDY     #MEMB-DTAB
9306  20429C               JSR     DSUBI           ; ADJUST 'BHIGH' FOR NEXT LINE.

                   ; ALLOCATE A BLOCK AT 'RTMP'.

9309  A252         :XR330  LDX     #MEMA-DTAB
930B  A019                 LDY     #RTMP-DTAB
930D  20459A               JSR     DMOVI

9310  20C19A               JSR     MALLOC          ; ALLOCATE IT (MUST BE ROOM).

9313  A4D4                 LDY     MEMB            ; COPY STATEMENT FROM 'TEXBUF'.
9315  88                   DEY                     ; # - 1 OF BYTES.

9316  B9008C       :XR350  LDA     TEXBUF,Y
9319  9199                 STA     (RTMP),Y
931B  88                   DEY
931C  10F8 ^9316           BPL     :XR350

                   ; STATEMENT HAS BEEN INSERTED.

931E  A219                 LDX     #RTMP-DTAB
9320  A054                 LDY     #MEMB-DTAB
9322  20329C               JSR     DADDI           ; ADJUST 'RTMP' FOR NEXT LINE.
```

```
                     ; ANY MORE TO MOVE?

9325  A213                  LDX     #BLOW-DTAB
9327  A015                  LDY     #BHIGH-DTAB
9329  20159C                JSR     DCMPI
932C  90AA ^92D8            BCC     :XR300          ; YES.

932E  202CB5         :XR500 JSR     RDYMES          ; ALL DONE.
9331  A900                  LDA     #0              ; SET CC FOR NORMAL EXIT.
9333  F029 ^935E            BEQ     :XR990          ; (BRA).

                     ; ERROR -- MAXIMUM LINE NUMBER EXCEEDED.

9335  A99B           :XR600 LDA     #RENERR         ; CAN'T RENUMBER.
9337  20FFB4                JSR     MESSOT
933A  A98D                  LDA     #LNCERR         ; LINE # OUT OF RANGE.
933C  20FFB4                JSR     MESSOT
933F  4C2E93                JMP     :XR500

                     ;
                     ; ERROR -- OVERLAPPING RANGE.
                     ;         'ALINE' FIRST NEW LINE.
                     ;         'RTMP'  LAST.

9342  A99B           :XR700 LDA     #RENERR         ; CAN'T RENUMBER.
9344  20FFB4                JSR     MESSOT
9347  A99C                  LDA     #OVLPER         ; OVERLAPPING RANGE.
9349  20FFB4                JSR     MESSOT
934C  A27A                  LDX     #ALINE-DTAB     ; FIRST NEW LINE.
934E  20149E                JSR     DECASC
9351  A99D                  LDA     #TOMES          ; TO.
9353  20FFB4                JSR     MESSOT
9356  A219                  LDX     #RTMP-DTAB      ; LAST NEW LINE.
9358  20149E                JSR     DECASC
935B  4C2E93                JMP     :XR500

                     ; ERROR -- SYNTAX.

935E                 :XR900

                     ; EXIT.

935E  08             :XR990 PHP                     ; SAVE CC.
935F  A4AB                  LDY     XTEMP           ; RESTORE Y.
9361  28                    PLP                     ; RESTORE CC.
9362  60                    RTS

                     ; DEFAULTS FOR 'RENUMBER'.

9363  0A00           RENNMS DW      10              ; FIRST NEW.
9365  0A00                  DW      10              ; INCREMENT.
9367  0000                  DW      0               ; FIRST OLD.
9369  0F27                  DW      MAXLN           ; LAST OLD.
936B  FFFF                  DW      EONMLS
```

```
 936D                         PROC
                    ;
                    ; MNYNMS -- RETURN 'MANY' NUMBERS FROM 'INLN'
                    ;
                    ; CALLING SEQUENCE:
                    ;
                    ;       'INLN' POINTS TO THE STATEMENT LINE
                    ;       Y       = CURRENT OFFSET IN 'INLN'
                    ;       'POINT' = LIST OF VALUES FOR INITIALIZING 'NMSBF'
                    ;
                    ;       JSR MNYNMS
                    ;       BNE     SYNTAX ERROR, RANGE ERROR, OR TOO MANY NUMBERS (A=ERROR
                    ;               CODE)
                    ;
                    ;       X       = NUMBER OF NUMBERS FOUND.
                    ;       'NMSBF' = LIST OF NUMBERS
                    ;       Y       = CURRENT OFFSET IN 'INLN'
                    ;       XTEMP   = CURRENT OFFSET IN 'INLN'
                    ;
                    ;       'NMSBF' IS INITIALIZED FROM THE LIST ADDRESSED BY 'POINT'.
                    ;       'EONMLS'  REPRESENTS THE END-OF-LIST.
                    ;       IF TOO MANY NUMER ARE IN THE SOURCE TEXT, AN ERROR CODE WILL BE
                    ;       RECOGNIZED.
                    ;
 936D  84AB        MNYNMS  STY     XTEMP
 936F  A000                LDY     #0              ; INITIALIZE 'NMSBF' FROM 'POINT'.

 9371  B186        :MN010  LDA     (POINT),Y
 9373  995305              STA     NMSBF,Y
 9376  C8                  INY
 9377  B186                LDA     (POINT),Y
 9379  995305              STA     NMSBF,Y
 937C  C8                  INY
 937D  C9FF                CMP     #HIGH EONMLS    ; CHECK FOR END OF LIST.
 937F  D0F0 ^9371          BNE     :MN010

 9381  A4AB                LDY     XTEMP           ; RESTORE Y.
 9383  A900                LDA     #0
 9385  8586                STA     POINT           ; INITIALIZE OFFSET TO STORE NEXT VALUE.

 9387  20079F      :MN020  JSR     SKPSEP          ; SKIP LOADING SEPARATORS.
 938A  206E81              JSR     ATOM            ; GET NEXT NUMBER.
 938D  D032 ^93C1          BNE     :MN099          ; ERROR -- RETURN.

 938F  C901                CMP     #NULL           ; CHECK FOR 'EOL'.
 9391  F02E ^93C1          BEQ     :MN099          ; EOL -- DONE.

 9393  C902                CMP     #NUM            ; CHECK FOR NUMBER.
 9395  D026 ^93BF          BNE     :MN090          ; NO -- ERROR.

 9397  84AB                STY     XTEMP           ; SAVE Y.
 9399  A238                LDX     #NUMBER-DTAB    ; CHECK IF NUMBER IS IN RANGE.
 939B  206B94              JSR     CHKLN
 939E  901B ^93BB          BCS     :MN080          ; NO -- OUT OF RANGE.

 93A0  A686                LDX     POINT           ; INDEX IN 'NMSBF'.
 93A2  A9FF                LDA     #$FF            ; CHECK IF TOO MANY VALUES.
```

```
93A4  DD5405              CMP     NMSBF+1,X
93A7  F016 ^93BF          BEQ     :MN090          ; YES -- TOO MANY.

93A9  A5B8                LDA     NUMBER          ; COPY TO NEXT POSITION IN 'NMSBF'.
93AB  9D5305              STA     NMSBF,X
93AE  A5B9                LDA     NUMBER+1
93B0  9D5405              STA     NMSBF+1,X
93B3  E8                  INX
93B4  E8                  INX
93B5  86B6                STX     POINT           ; UPDATE INDEX.

93B7  A4AB                LDY     XTEMP           ; RESTORE Y.
93B9  D0CC ^9387          BNE     :MN020          ; (BRA).

                  ; OUT OF RANGE

93BB  A98D        :MN080  LDA     #LNOERR
93BD  D002 ^93C1          BNE     :MN099          ; (BRA).

                  ; NEXT 'ATOM' IS NOT A NUMBER, OR TOO MANY NUMBERS.

93BF  A902        :MN090  LDA     #IMPERR         ; IMPROPER PARAMETER ERROR.

93C1  08          :MN099  PHP                     ; SAVE CC.
93C2  A5B6                LDA     POINT           ; INDEX IN 'NMSBF'.
93C4  4A                  LSR     A
93C5  AA                  TAX                     ; AS ADVERTISED.
93C6  28                  PLP
93C7  60                  RTS



93C8                      PROC
                  ;
                  ; BRACKT -- BRACKET A RANGE OF LINES
                  ;
                  ; CALLING SEQUENCE:
                  ;
                  ;       'LS'    = START OF RANGE
                  ;       'LEND'  = END
                  ;
                  ;       JSR     BRACKT
                  ;       BNE     'LS' > 'LEND'
                  ;
                  ;       'BLOW'  = ADDRESS OF START OF RANGE
                  ;       'BHIGH' = ADDRESS OF (PAST) END OF RANGE
                  ;       'BNUM'  = # OF LINES IN THE RANGE
                  ;
                  ;       USES 'LINENO', 'POINT'
                  ;
93C8  A260        BRACKT  LDX     #LEND-DTAB      ; CHECK IF 'LS' <= 'LEND'
93CA  A05E                LDY     #LS-DTAB
93CC  20159C              JSR     DCMPI
93CF  9040 ^9411          BCC     :BR090          ; ERROR -- 'LS' > 'LEND'

93D1  A900                LDA     #0              ; INITIALIZE # OF LINES.
```

```
93D3  8597                STA     BNUM
93D5  8598                STA     BNUM+1

93D7  A25E                LDX     #LS-DTAB
93D9  205D94              JSR     LNFIND

93DC  A213                LDX     #BLOW-DTAB       ; 'BLOW' = ADDRESS OF 'LS' OR SUCCESSOR.
93DE  A04E                LDY     #PP-DTAB
93E0  20459A              JSR     DMOVI
93E3  A236                LDX     #POINT-DTAB      ; USE 'POINT' FOR CURRENT LINE ADDRESS.
93E5  20459A              JSR     DMOVI

93E8               :BR010
                   ; *S*  LDX     #POINT-DTAB
93E8  20139A              JSR     SEND             ; CHECK IF END OF LIST.
93EB  F01B ^9408          BEQ     :BR050           ; YES -- DONE.

93ED  208C9F              JSR     GTLNNO           ; 'LINENO' IN L/H ORDER FROM 'POINT'.
93F0  A260                LDX     #LEND-DTAB       ; CHECK IF CURRENT LINE IS IN RANGE.
93F2  A05C                LDY     #LINENO-DTAB
93F4  20159C              JSR     DCMPI
93F7  900F ^9408          BCC     :BR050           ; NO -- NOT IN RANGE.

93F9  A901                LDA     #1               ; ONE MORE LINE IN RANGE.
93FB  A217                LDX     #BNUM-DTAB
93FD  20049D              JSR     DADDS

9400  A236                LDX     #POINT-DTAB      ; POINT TO NEXT LINE.
9402  20AA9A              JSR     SNXTI
9405  4CE893              JMP     :BR010           ; CHECK NEXT LINE.

                   ; CURRENT LINE IS NOT IN THE RANGE.

9408  A215         :BR050 LDX     #BHIGH-DTAB      ; AS ADVERTISED.
940A  A036                LDY     #POINT-DTAB
940C  20459A              JSR     DMOVI

940F  A900                LDA     #0               ; SET CC FOR EXIT.
                   ; *S*  RTS

                   ; ERROR -- 'LS' > 'LEND'

9411  60           :BR090 RTS


9412                      PROC
                   ;
                   ; GTLSLN -- GET LINE NUMBER OF LAST PROGRAM LINE + 10
                   ;
                   ; CALLING SEQUENCE:
                   ;
                   ;        JSR     GTLSLN
                   ;
                   ;        'LINENO' = LAST LINE NUMBER + 10 (0 IF EMPTY).
                   ;
9412  84[illegible]       GTLSLN STY     XTEMP            ; SAVE Y.
```

```
9414  A900                      LDA     #0                 ; 'EMPTY' VALUE.
9416  850C                      STA     LINENO
9418  850D                      STA     LINENO+1
941A  209F9E                    JSR     STMLST             ; 'LP' = 'S1L'

941D  A23A                      LDX     #LP-DTAB
941F  20139A                    JSR     SEND               ; TRAP FOR PROGRAM EMPTY.
9422  F014 ^9438                BEQ     :GL090             ; EMPTY.

9424  A03A            :GL010    LDY     #LP-DTAB           ; UPDATE 'POINT'.
9426  A236                      LDX     #POINT-DTAB
9428  20459A                    JSR     DMOVI

942B  A23A                      LDX     #LP-DTAB           ; NEXT LINE.
942D  20AA9A                    JSR     SNXTI
9430  20139A                    JSR     SEND
9433  D0EF ^9424                BNE     :GL010             ; KEEP CHECKING.

9435  208C9F                    JSR     GTLNNO             ; 'LINENO' FROM 'POINT'.

9438  A25C            :GL090    LDX     #LINENO-DTAB       ; LAST LINE + 10
943A  A90A                      LDA     #10
943C  20049D                    JSR     DADDS

943F  A4AB                      LDY     XTEMP
9441  60                        RTS                        ; RESTORE Y.



9442                            PROC
                      ;
                      ; RENFND -- FIND LINE FOR 'RENUMBER'
                      ;
                      ; CALLING SEQUENCE:
                      ;
                      ;         X        = DTAB OFFSET TO LINE NUMBER
                      ;         'BLOW'   = ADDRESS OF FIRST LINE TO RENUMBER
                      ;         'BHIGH'  = ADDRESS PAST LAST
                      ;
                      ;         JSR      RENFND
                      ;         A        = 0 VALID
                      ;                    1 LINE OVERLAPS A NON-RENUMBERED ONE (ERROR)
                      ;
                      ;         PP       = ADDRESS OF LINE (OR SUCCESSOR)
                      ;
9442  205094          RENFND    JSR     LNFIND             ; FIND ADDRESS.
9445  D010 ^9457                BNE     :RFD80             ; NO OVERLAP.

9447  A24E                      LDX     #PP-DTAB           ; >= 'BLOW'?
9449  A013                      LDY     #BLOW-DTAB
944B  20159C                    JSR     DCMPI
944E  900A ^945A                BCC     :RFD90             ; NO -- ERROR.

                      ; *S*     LDX     #PP-DTAB
9450  A015                      LDY     #BHIGH-DTAB        ; < 'BHIGH'?
9452  20159C                    JSR     DCMPI
9455  B003 ^945A                BCS     :RFD90             ; NO -- ERROR.
```

```
9457  A900          :RFD80  LDA     #0                  ; CLEAR A FOR EXIT.
9459  60                    RTS

945A  A901          :RFD90  LDA     #1                  ; SET A FOR ERROR.
945C  60                    RTS


945D                        PROC
                    ;
                    ; LNFIND -- FIND LINE NUMBER
                    ;
                    ; CALLING SEQUENCE
                    ;
                    ;       X  = OFFSET TO DTAB LINE NUMBER
                    ;
                    ;       JSR LNFIND
                    ;       BNE NOT FOUND (PP POINTS TO SUCCESSOR)
                    ;
                    ;       PP = ADDRESS OF LINE (OR SUCCESSOR)
                    ;
                    ; USES LINENO
                    ;
945D  B580          LNFIND  LDA     DTAB,X              ; INVERT LINE NUMBER FOR SEARCH.
945F  85DD                  STA     LINENO+1
9461  B581                  LDA     DTAB+1,X
9463  85DC                  STA     LINENO
9465  20ED7A                JSR     NUMNAM              ; SETUP 'LINENO' FOR SEARCH.
9468  4C5B99                JMP     IFIND               ; FIND LINE (OR SUCCESSOR).


946B                        PROC
                    ; CHKLN -- CHECK STATEMENT LINE # FOR OUT OF RANGE.
                    ;
                    ; CALLING SEQUENCE:
                    ;
                    ;       X = DTAB INDEX TO LINE NUMBER.
                    ;
                    ;       JSR     CHKLN
                    ;       BCS     OUT OF RANGE (A = ERROR CODE)
                    ;
946B  A027          CHKLN   LDY     # HIGH [MAXLN+1]
946D  A910                  LDA     # LOW [MAXLN+1]
946F  200F9C                JSR     DCWCI
9472  9002 ^9476            BCC     :CL090              ; NOT OUT OF RANGE.

9474  A98E                  LDA     #LNOERR

9476  60            :CL090  RTS


9477                        PROC
                    ;
```

```
                        ; NMOVI -- MOVE VALUE FROM 'NMSBF'
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       X = DTAB OFFSET TO DESTINATION
                        ;       Y = 'NMSBF' OFFSET
                        ;
                        ;       JSR NMOVI
                        ;
                        ;       DTAB(X) = NMSBF+Y,+Y+1
                        ;
9477  B95305            NMOVI   LDA     NMSBF,Y
947A  9580                      STA     DTAB,X
947C  B95405                    LDA     NMSBF+1,Y
947F  9581                      STA     DTAB+1,X
9481  60                        RTS
```

```
                          ;
                          ; I/O SUBSYTEM ROUTINES
                          ;


 9482                             PROC
                          ;
                          ; CHOT -- PRINT ONE CHARACTER TO "E:".
                          ;
                          ;       A = ATASCII CHARACTER
                          ;       'CDEST' = I/O ROUTINE OFFSET OR $80+XX OR $FF.
                          ;
                          ;       JSR     CHOT
                          ;
 9482  86A1               CHOT    STX     TEMP            ; SAVE REGISTERS.
 9484  84A2                       STY     TEMP+1

 9486  AE3005                     LDX     CDEST           ; PREPARE TO OUTPUT TO DEVICE.
 9489  300E ^9499                 BMI     :CH100          ; SPECIAL OUTPUT.

 948B  20B797                     JSR     IOHAND

                          ; *** EXTERNAL ENTRY POINT ***

 948E  C000               IOERCK  CPY     #0              ; ERROR CHECK.
 9490  101A ^94AC                 BPL     :CH120          ; O.K.

                          ; *** EXTERNAL ENTRY POINT ***

 9492  84E4               IOE010  STY     IOSTAT          ; SAVE I/O STATUS.
 9494  A986                       LDA     #IOERR
 9496  4C3A7A                     JMP     PSTOP           ; STOP ON ERROR.

 9499  E0FF               :CH100  CPX     #$FF            ; RESULT TO 'TEXBUF'?
 949B  F005 ^94A2                 BEQ     :CH110          ; YES.

 949D  A230                       LDX     #IOCB3          ; NO -- TO IOCB 3.
 949F  4C6397                     JMP     DIO005          ; OUTPUT CHARACTER AND RETURN.

 94A2  A48F               :CH110  LDY     TELN+3          ; GET INDEX.
 94A4  C0FE                       CPY     #TEXLNG         ; BUFFER FULL?
 94A6  F004 ^94AC                 BEQ     :CH120          ; YES -- STORE NO MORE.

 94A8  918C                       STA     (TELN),Y        ; NO -- STORE CHARACTER.
 94AA  E68F                       INC     TELN+3

 94AC  A4A2               :CH120  LDY     TEMP+1          ; RESTORE REGISTERS.
 94AE  A6A1                       LDX     TEMP
 94B0  60                         RTS



 94B1                             PROC
                          ;
                          ; GETLIN -- GET LINE FROM "E:"
                          ;
```

```
                        ; CALLING SEQUENCE:
                        ;
                        ;       X = OFFSET TO BUFFER ADDRESS.
                        ;
                        ;       JSR     GETLIN
                        ;
                        ;       DTAB(X+2) = 0 -- START INDEX.
                        ;       DTAB(X+3) = LINE LENGTH -- END INDEX.
                        ;
94B1  86A3              GETLIN  STX     TEMP+2          ; SAVE INDICES.
94B3  84A4                      STY     TEMP+3

94B5  A900                      LDA     #0              ; ENABLE TEXT CURSOR.
94B7  95B2                      STA     DTAB+2,X        ; AS ADVERTISED.
94B9  20A79F                    JSR     CRSNOP          ; MAKE CURSOR SHOW NOW.

94BC  B5B0                      LDA     DTAB,X          ; SETUP BUFFER ADDRESS.
94BE  8D4403                    STA     IOCB0+ICBAL
94C1  B5B1                      LDA     DTAB+1,X
94C3  8D4503                    STA     IOCB0+ICBAH

94C6  A905                      LDA     #GETR           ; GET RECORD COMMAND.
94C8  8D4203                    STA     IOCB0+ICCOM

94CB  A979              :GL010  LDA     #LINLNG-1       ; SETUP MAXIMUM LINE LENGTH FOR READ.
94CD  8D4803                    STA     IOCB0+ICBLL

94D0  A200                      LDX     #IOCB0          ; IOCB 0.
94D2  8E4903                    STX     IOCB0+ICBLH     ; *S*.
94D5  2056E4                    JSR     CIO             ; DO I/O.

94D8  C089                      CPY     #$89            ; TRUNCATED RECORD?
94DA  D008 ^94E4                BNE     :GL020          ; NO.

94DC  A90F                      LDA     #OLLERR         ; YES -- INFORM OPERATOR & TRY AGAIN.
94DE  20FFB4                    JSR     MESSOT
94E1  4CCB94                    JMP     :GL010

94E4  98                :GL020  TYA                     ; ERROR CHECK
94E5  30AB ^9492                BMI     ICE010          ; ERROR.

94E7  20A79F                    JSR     CRSNOP          ; DISABLE TEXT CURSOR (A <> 0).

94EA  A6A3                      LDX     TEMP+2          ; RESTORE INDICES.
94EC  A4A4                      LDY     TEMP+3
94EE  AD4803                    LDA     IOCB0+ICBLL     ; SETUP END INDEX.
94F1  95B3                      STA     DTAB+3,X
94F3  60                        RTS

94F4                            PROC
                        ;
                        ; TXOPEN -- OPEN THE TEXT SCREEN.
                        ;
94F4  A200              TXOPEN  LDX     #0              ; RESET GRAPHICS MODE FLAG.
94F6  8E1405                    STX     GRFLAG
94F9  8E5105                    STX     LETTRSZ         ; SMALL LETTERS.
```

```
  94FC  8E4F05              STX     TRTLON          ; VISIBLE TURTLE OFF.
  94FF  200E46              JSR     TRONOF

  9502  202996              JSR     COMPRS          ; COMPRESS MEMORY.
  9505  A220                LDX     #IOCB2          ; CLOSE 'S'.
  9507  203F97              JSR     DCLOSE
  950A  206E96              JSR     EOPEN           ; OPEN 'E'.
  950D  4C5C96              JMP     EXPAND          ; EXPAND MEMORY & RETURN.


  9510                      PROC
                    ;
                    ; GSOPEN -- OPEN THE GRAPHICS SCREEN
                    ;
                    ; THIS ROUTINE COMPRESSES MEMORY, OPENS THE GRAPHICS SCREEN AND DE-COMPRESSES
                    ; THE MEMORY AGAIN.
                    ;
                    ;
                    ; CALLING SEQUENCE:
                    ;
                    ;       'GSMODE'   = SCREEN MODE NUMBER.
                    ;       'SPLTSC'   = SPLIT SCREEN OPTION SELECT.
                    ;       'FINEFG'   = FINE SCROLLING FLAG.
                    ;
                    ;       JSR     GSOPEN
                    ;
                    ;       'XC' & 'YC'= SCREEN CENTER.
                    ;       'S2L'&'S2H'= VAR LIMITS.
                    ;       'APPMHI'   = TOP OF VARIABLES.
                    ;       'TRTLON'   = 1 OR 0.
                    ;       'THETA'    = 0.
                    ;       'GX' & 'GY'= 0.
                    ;
  9510  202996      GSOPEN  JSR     COMPRS          ; COMPRESS MEMORY.


                    ; NOW ATTEMPT TO OPEN 'S:' TO THE DESIRED SCREEN MODE; THERE MAY NOT BE
                    ; ENOUGH MEMORY, HOWEVER.

  9513  A220        :G0010  LDX     #IOCB2
  9515  203F97              JSR     DCLOSE

  9518  A953                LDA     #'S'            ; DEVICE NAME = 'S'.
  951A  8D2005              STA     OPNBUF
  951D  A99B                LDA     #EOL
  951F  8D2105              STA     OPNBUF+1
  9522  A90C                LDA     #OWRIT+OREAD    ; SCREEN OPTIONS.
  9524  0D5205              ORA     SPLTSC
  9527  AE3705              LDX     GSMODE          ; IF NO MODE CHANGE ...
  952A  E457                CPX     DINDEX
  952C  D002 ^9530          BNE     :G0012

  952E  0920                ORA     #NOCLR          ; ... THEN DON'T CLEAR SCREEN.

  9530  8D6A03      :G0012  STA     IOCB2+ICAUX1
  9533  8E6B03              STX     IOCB2+ICAUX2
  9536  A903                LDA     #OPEN           ; OPEN COMMAND.
```

```
9538  8D6203              STA    IOCB2+ICCOM

953B  A220                LDX    #IOCB2          ; OPEN DEVICE ON IOCB 2.
953D  208C97              JSR    BUFPNT          ; SETUP OPEN BUFFER POINTER.
9540  2056E4              JSR    CIO
9543  84E4                STY    IOSTAT          ; SAVE STATUS FOR LATER.
9545  1003 ^954A          BPL    :G0013
9547  4C1196              JMP    :G0020          ; ERROR -- DON'T PLOT POINT.

954A  CEF002      :G0013  DEC    CRSINH          ; INHIBIT THE CURSOR.

954D  AD3705              LDA    GSMODE          ; SETUP MODE DEPENDENT VARIABLES
9550  0A                  ASL    A               ; X2.
9551  AA                  TAX
9552  BD16B8              LDA    XCENTR,X
9555  8D6105              STA    XC
9558  BD17B8              LDA    XCENTR+1,X
955B  8D6205              STA    XC+1
955E  BD36B8              LDA    YCENTR,X
9561  8D6305              STA    YC
9564  BD37B8              LDA    YCENTR+1,X
9567  8D6405              STA    YC+1

956A  BD56B8              LDA    COLMAX,X        ; SET 'FLOOD' LIMITS.
956D  8DAC05              STA    MAXCOL
9570  BD57B8              LDA    COLMAX+1,X
9573  8DAD05              STA    MAXCOL+1
9576  BD76B8              LDA    ROWMAX,X
9579  8DAB05              STA    MAXROW

957C  AE3705              LDX    GSMODE
957F  BD06B8              LDA    COLRS,X         ; # OF FOREGROUND COLORS.
9582  8DB905              STA    NCOLRS
9585  E003                CPX    #2+1            ; SEE IF MODES 1 OR 2.
9587  B022 ^95AB          BCS    :G0015          ; NO -- MODE 3-15

9589  BD96B8              LDA    LMRGTB,X        ; SET MARGINS FOR LARGE LETTERS.
958C  8DB505              STA    LFCOL
958F  BD99B8              LDA    RMRGTB,X
9592  8DB605              STA    RGCOL
9595  18                  CLC
9596  ADAB05              LDA    MAXROW
9599  6902                ADC    #2              ; SET 'BOTSCR' FOR 'XPOS'.
959B  8DBF02              STA    BOTSCR
959E  A900                LDA    #KOFF           ; SET TURTLE OFF.
95A0  8D4F05              STA    TRTLON
95A3  A916                LDA    #SPUTC-IOVBAS   ; ROUTE OUTPUTS TO S:.
95A5  8D3005              STA    CDEST
95A8  4C1196              JMP    :G0020          ; AVOID TURTLE SETUP.

95AB  AD6A03      :G0015  LDA    IOCB2+ICAUX1    ; WAS SCREEN CLEARED?
95AE  2920                AND    #NOCLR
95B0  F013 ^95C5          BEQ    :G0017          ; YES.

95B2  ADB905              LDA    NCOLRS          ; NO -- RE-ESTABLISH COLOR REGS.

95B5  48          :G0016  PHA
```

```
95B6  AA                      TAX
95B7  BD9B05                  LDA     PNCLRS,X
95BA  20F7A4                  JSR     SETCLR
95BD  68                      PLA
95BE  38                      SEC
95BF  E901                    SBC     #1
95C1  10F2 ^95B5              BPL     :G0016

95C3  3016 ^95DB              BMI     :G0018             ; (BRA).

95C5  2075AF        :G0017    JSR     DFCLRS             ; SET DEFAULT COLORS FOR MODE.
95C8  A906                    LDA     #EPUTC-IOVBAS      ; ROUTE OUTPUTS TO E:.
95CA  8D3005                  STA     CDEST
95CD  A900                    LDA     #0                 ; CLEAR WALL SELECTION.
95CF  8DCD05                  STA     WALLS
95D2  8DCE05                  STA     WALLS+1
95D5  8D1305                  STA     PEN                ; SET PEN TO ERASE & DOWN.
95D8  8D5105                  STA     LETTRSZ            ; LETTER SIZE = SMALL.

95DB  A901          :G0018    LDA     #KON               ; SET TURTLE ON.
95DD  8D4F05                  STA     TRTLON

95E0  20B7A3                  JSR     GHOME              ; TURTLE HOME.
95E3  20CDA3                  JSR     GNORTH             ; TURTLE NORTH.
95E6  AD5205                  LDA     SPLTSC             ; SPLIT SCREEN MODE?
95E9  F026 ^9611              BEQ     :G0020             ; NO -- FULL SCREEN.

95EB  A9E7                    LDA     # LOW GRDLI        ; SETUP PILOT'S DLI VECTOR.
95ED  8D0002                  STA     VDSLST
95F0  A9B4                    LDA     # HIGH GRDLI
95F2  8D0102                  STA     VDSLST+1

95F5  A9C0                    LDA     #$C0               ; ENABLE VBLANK & DLI.
95F7  8D0ED4                  STA     NMIEN

95FA  AE3705                  LDX     GSMODE             ; GET MODE DEPENDENT OFFSET FROM START ...
95FD  BDD088                  LDA     DLIOFF,X           ; ... OF DISPLAY LIST TO LOC OF DLI.
9600  A8                      TAY
9601  AD3002                  LDA     SDLSTL
9604  85A1                    STA     TEMP
9606  AD3102                  LDA     SDLSTL+1
9609  85A2                    STA     TEMP+1
960B  B1A1                    LDA     (TEMP),Y           ; SET THE DLI BIT.
960D  0980                    ORA     #$8
960F  91A1                    STA     (TEMP),Y

9611  200CA6        :G0020    JSR     TRONOF             ; ENABLE OR DISABLE VISIBLE TURTLE.
9614  205C96                  JSR     EXPAND             ; EXPAND MEMORY.

9617  A4E4                    LDY     IOSTAT             ; SEE IF THERE WAS AN I/O ERROR.
9619  1008 ^9623              BPL     :G0090             ; NO.

961B  A220                    LDX     #IOCB2             ; YES -- CLOSE DEVICE & REPORT ERROR.
961D  203F97                  JSR     DCLOSE
9620  4C9294                  JMP     IOE010

9623  A901          :G0090    LDA     #1
```

```
9625  8D1405           STA    GRFLAG        ; SET GRAPHICS SCREEN FLAG.
9628  60               RTS

9629                   PROC
                ; FIRST COMPRESS THE RAM STORAGE, LEAVING THE FREE AREA AT THE HIGH ADDRESSES
                ; BY REMOVING THE GAP BETWEEN THE PROGRAM STORAGE AREA AND THE STRING
                ; STORAGE AREA.

9629  A580      COMPRS LDA    S1H           ; 'MDP' = 'S1H' (DESTINATION).
962B  85D8             STA    MDP
962D  A581             LDA    S1H+1
962F  85D9             STA    MDP+1

9631  A582             LDA    S2L           ; 'MSP' = 'S2L' (SOURCE).
9633  85D6             STA    MSP
9635  A583             LDA    S2L+1
9637  85D7             STA    MSP+1

9639  38               SEC                  ; 'MBC' = 'S2H' - 'S2L' (BYTE COUNT).
963A  A584             LDA    S2H           ; ('CETEMP' = SAME).
963C  E582             SBC    S2L
963E  85DA             STA    MBC
9640  8DB305           STA    CETEMP        ; (SAVE FOR LATER).
9643  A585             LDA    S2H+1
9645  E583             SBC    S2L+1
9647  85DB             STA    MBC+1
9649  8DB405           STA    CETEMP+1

964C  18               CLC                  ; 'APPMHI' = 'S1H' + 'MBC'.
964D  A580             LDA    S1H
964F  65DA             ADC    MBC
9651  85DE             STA    APPMHI
9653  A581             LDA    S1H+1
9655  65DB             ADC    MBC+1
9657  85DF             STA    APPMHI+1

9659  4CA69B           JMP    MOVIA         ; MOVE STRING STORAGE DOWN & RETURN.

965C                   PROC
                ; NOW MOVE THE STRING STORAGE AREA UP TO THE CURRENT TOP OF MEMORY SO
                ; THAT THE FREE AREA IS ONCE AGAIN BETWEEN THE PROGRAM STORAGE AREA AND
                ; THE STRING STORAGE AREA.

965C  ADB305    EXPAND LDA    CETEMP        ; 'MBC' = PRIOR 'MBC' (BYTE COUNT).
965F  85DA             STA    MBC
9661  ADB405           LDA    CETEMP+1
9664  85DB             STA    MBC+1

9666  A580             LDA    S1H           ; 'MSP' = 'S1H' (SOURCE).
9668  85D6             STA    MSP
966A  A581             LDA    S1H+1
966C  85D7             STA    MSP+1

966E  38               SEC                  ; 'MDP' = 'MEMHI' - 'MBC' (DESTINATION).
966F  ADE502           LDA    MEMHI
9672  8584             STA    S2H           ; 'S2H' = 'MEMHI'.
9674  E5DA             SBC    MBC
```

```
9676  8508            STA     MDP
9678  8502            STA     S2L             ; 'S2L' = SAME AS NEW 'MDP'.

967A  ADE602          LDA     MEMHI+1         ; NOW AS ABOVE FOR MSB.
967D  8505            STA     S2H+1
967F  E50B            SBC     MSC+1
9681  8509            STA     MDP+1
9683  8503            STA     S2L+1

9685  A900            LDA     #0              ; ALLOWS RESET IN ANY MODE.
9687  850E            STA     APPMHI
9689  850F            STA     APPMHI+1

968B  4CC496          JMP     MOVDA           ; MOVE STRING STORAGE TO TOP OF MEM & RETURN.


968E                  PROC
               ;
               ; EOPEN -- OPEN IOCB 0 TO E:
               ;
968E  A200     EOPEN  LDX     #IOCB0          ; CLOSE E:
9690  203F97          JSR     DCLOSE

9693  ADB205          LDA     FINEFG          ; SET SCROLL MODE.
9696  8D6E02          STA     FINE

9699  A906            LDA     #EPUTC-IOVBAS   ; 'CHOT' OUTPUT TO E:.
969B  8D3005          STA     CDEST

969E  A945            LDA     #'E'            ; SET DEVICE NAME TO 'E'.
96A0  8D2005          STA     OPNBUF
96A3  A99B            LDA     #EOL
96A5  8D2105          STA     OPNBUF+1
96A8  A90C            LDA     #OREAD+OWRIT    ; OPEN 'E' AGAIN & RETURN.
96AA  20F496          JSR     DOPEN

96AD  A552            LDA     LMARGN          ; ESTABLISH MARGINS.
96AF  8DB505          STA     LFCOL
96B2  A553            LDA     RMARGN
96B4  8DB605          STA     RGCOL
96B7  CEF002          DEC     CRSINH          ; INHIBIT CURSOR.
96BA  60              RTS

96BB                  PROC
               ;
               ; TSTMOD -- TEST SCREEN MODE
               ;
               ; CALLING SEQUENCE:
               ;
               ;       GRFLAG  = 0 FOR TEXT, ELSE GRAPHICS.
               ;       SPLTSC  = 0 FOR FULL SCREEN, ELSE SPLIT.
               ;       LETTRS  = 0 FOR SMALL, ELSE MEDIUM OR LARGE.
               ;
               ;       JSR     TSTMOD
               ;
```

```
                       ;        A = 1 IF TEXT SCREEN, SMALL LETTERS,
                       ;            2 IF TEXT SCREEN, MEDIUM OR LARGE LETTERS,
                       ;            4 IF GRAPHICS SCREEN, WITH TEXT WINDOW (SPLIT),
                       ;            8 IF FULL GRAPHICS SCREEN.
                       ;
96BB  AD1405           TSTMOD  LDA     GRFLAG          ; GRAPHICS MODE?
96BE  D00B ^96CB               BNE     :TM030          ; YES.

96C0  AD5105                   LDA     LETTRSZ         ; NO -- CHECK FOR LETTER SIZE.
96C3  D003 ^96C8               BNE     :TM020          ; NOT SMALL.

96C5  A901                     LDA     #TXSL           ; SMALL.
96C7  60                       RTS

96C8  A902             :TM020  LDA     #TXML           ; MEDIUM OR LARGE.
96CA  60                       RTS

96CB  AD5205           :TM030  LDA     SPLTSC          ; SPLIT SCREEN GRAPHICS?
96CE  F003 ^96D3               BEQ     :TM040          ; NO -- FULL.

96D0  A904                     LDA     #GRSS           ; YES -- SPLIT SCREEN.
96D2  60                       RTS

96D3  A908             :TM040  LDA     #GRFS           ; FULL SCREEN GRAPHICS.
96D5  60                       RTS

96D6                           PROC
                       ;
                       ; DNAME -- EXTRACT DEVICE/FILENAME
                       ;
                       ; CALLING SEQUENCE:
                       ;
                       ;        Y = INDEX TO START OF NAME
                       ;
                       ;        JSR     DNAME
                       ;
                       ;        Y = INDEX TO NAME DELIMITER.
                       ;        X = 'OPNBUF' INDEX TO CHAR AFTER NAME (EOL).
                       ;        'OPNBUF' RECEIVES NAME.
                       ;
96D6  20139F           DNAME   JSR     SLB             ; SKIP LEADING BLANKS.
96D9  4CDC96                   JMP     FNAME           ; NAME TO 'OPNBUF' & RETURN.


96DC                           PROC
96DC  A200             FNAME   LDX     #0

96DE  B180             :FN010  LDA     (INLN),Y
96E0  20E09E                   JSR     CHKSEP          ; SEPARATOR?
96E3  F009 ^96EE               BEQ     :FN020          ; YES -- DONE.

96E5  9D2005                   STA     OPNBUF,X        ; NO -- PART OF NAME.
96E8  E8                       INX

96E9  C8                       INY
96EA  C00F                     CPY     #DNSIZE         ; NAME TOO LONG?
```

```
96EC  D0F0 ^96DE          BNE     :FN010          ; NO -- KEEP SCANNING.

96EE  A99B        :FN020  LDA     #EOL            ; APPEND EOL AFTER NAME.
96F0  9D2005              STA     OPNBUF,X
96F3  60                  RTS


96F4                      PROC
                  ;
                  ; DOPEN -- DEVICE OPEN
                  ;
                  ; CALLING SEQUENCE:
                  ;
                  ;       'IOEDIS' <> 0 INDICATES TO IGNORE I/O ERROR.
                  ;       X = IOCB INDEX.
                  ;       A = OPEN DIRECTION + AUX1 OPTIONS.
                  ;       'OPNBUF' CONTAINS DEVICE/FILENAME.
                  ;
                  ;       JSR     DOPEN
                  ;
                  ; RETURNS ONLY IF OPEN SUCCEEDED.
                  ;
96F4  48          DOPEN   PHA                     ; SAVE OPEN CODE.
96F5  203F97              JSR     DCLOSE          ; *** JUST IN CASE ***.
96F8  68                  PLA                     ; RESTORE OPEN CODE.
96F9  86A1                STX     TEMP
96FB  84A2                STY     TEMP+1
96FD  0D1D05              ORA     AUX1            ; MERGE USER BYTE.
9700  9D4A03              STA     ICAUX1,X        ; SETUP OPEN DIRECTION.

9703  20B798              JSR     CHKDEV          ; CHECK FOR INVALID OPEN.

9706  AD1E05              LDA     AUX2            ; SETUP AUX2.
9709  9D4B03              STA     ICAUX2,X

970C  A900                LDA     #0
970E  9D4803              STA     ICBLL,X         ; SETUP FOR ACCUMULATOR XFER OF DATA.
9711  9D4903              STA     ICBLH,X
9714  8D1D05              STA     AUX1            ; CLEAR USER BYTES.
9717  8D1E05              STA     AUX2

971A  A903                LDA     #OPEN           ; OPEN COMMAND.
971C  9D4203              STA     ICCOM,X

971F  208C97              JSR     BUFPNT          ; SETUP OPEN BUFFER POINTER.

9722  2056E4              JSR     CIO
                  ;       JSR     COLORS          ; RE-ESTABLISH SPECIAL COLORS.
                  ;                               ; *** NEEDED ONLY IF OUTPUT TO S: OR E: ALLOWED
                  ;                               ;     IN GRAPHICS MODE ***

9725  98                  TYA                     ; CHECK STATUS.
9726  1026 ^974E          BPL     DOP010          ; O.K.

                  ; *** EXTERNAL ENTRY POINT ***
```

```
                            ;
                            ;         X = IOCB INDEX.
                            ;         Y = ERROR STATUS ON ENTRY.

9728  AD0405        DOP005   LDA      IOEDIS           ; ERROR STOP DISABLED?
972B  08                     PHP
972C  A99B                   LDA      #EOL             ; (RETURN EOL CHAR ON ERROR).
972E  28                     PLP
972F  F008 ^9739             BEQ      :D0007           ; NO.

9731  A5FF                   LDA      RUN              ; YES -- IS IT ALSO RUN MODE?
9733  08                     PHP
9734  A99B                   LDA      #EOL             ; RETURN EOL ON ERROR.
9736  28                     PLP
9737  D015 ^974E             BNE      DOP010           ; YES.

9739  203F97        :D0007   JSR      DCLOSE           ; NO -- CLOSE FILE IN ERROR.
973C  4C9294                 JMP      IOE010           ; ERROR -- STOP (SKIP BRANCH POINT).


973F                         PROC
                            ;
                            ; DCLOSE -- CLOSE IOCB
                            ;
                            ; CALLING SEQUENCE:
                            ;
                            ;         X = IOCB INDEX
                            ;
                            ;         JSR     DCLOSE
                            ;
                            ; NOTE: CLOSE STATUS IS OF NO IMPORTANCE TO THIS ROUTINE.
                            ;
973F  86A1          DCLOSE   STX      TEMP
9741  84A2                   STY      TEMP+1
9743  A90C                   LDA      #CLOSE
9745  9D4203                 STA      ICCOM,X

9748  2056E4                 JSR      CIO

974B  20B49F                 JSR      AUDCLR           ; CLEAR AUDIO REGISTERS.

                            ; *** EXTERNAL ENTRY POINT ***

974E                DOP010
974E  A6A1          DIO010   LDX      TEMP             ; RESTORE REGISTERS.
9750  A4A2                   LDY      TEMP+1
9752  60                     RTS


9753                         PROC
                            ;
                            ; DIN & DOUT -- IOCB DATA IN AND OUT
                            ;
                            ; CALLING SEQUENCE:
                            ;
```

```
                        ;        'IOEDIS' <> 0 INDICATES TO IGNORE I/O ERROR.
                        ;        X = IOCB INDEX
                        ;        A = DATA ('DOUT' ONLY)
                        ;
                        ;        JSR     DIN/DOUT
                        ;
                        ;        A = DATA ('DIN' ONLY), RETURNS EOL ON ERROR.
                        ;
9753  48                DIN      PHA
9754  A907                       LDA     #GETC           ; SETUP COMMAND BYTE.
9756  D003 ^975B                 BNE     :I0003          ; (BRA).


9758  48                DOUT     PHA                     ; SAVE DATA BYTE.
9759  A908                       LDA     #PUTC           ; SETUP COMMAND BYTE.

975B  9D4203            :I0003   STA     ICCOM,X
975E  68                         PLA
975F  86A1                       STX     TEMP            ; SAVE REGISTERS.
9761  84A2                       STY     TEMP+1

                        ; *** EXTERNAL ENTRY POINT FROM 'CHOT' ***

9763  2056E4            DI0005   JSR     CIO

9766  84E4                       STY     IOSTAT          ; SAVE I/O STATUS.
9768  C000                       CPY     #0              ; CHECK STATUS.
976A  10E2 ^974E                 BPL     DI0010          ; O.K.

976C  A99B                       LDA     #EOL            ; RETURN EOL ON ERROR.
976E  C088                       CPY     #$88            ; END OF FILE?
9770  D0B6 ^9728                 BNE     DOP005          ; NO -- FATAL ERROR (SKIP BRANCH).

9772  F0DA ^974E                 BEQ     DI0010          ; YES -- RETURN EOL (BRA).


9774                             PROC
                        ;
                        ; KIN -- KEYBOARD CHARACTER INPUT
                        ;
                        ; CALLING SEGUENCE:
                        ;
                        ;        JSR     KIN
                        ;
                        ;        A =     ATASCII CHAR
                        ;
9774  86A1              KIN      STX     TEMP            ; SAME REGISTERS.
9776  84A2                       STY     TEMP+1
9778  A224                       LDX     #KGETC-IOVBAS   ; GET CHAR FROM 'K'.
977A  20B797                     JSR     IOHAND
977D  4C8E94                     JMP     IOERCK          ; CHECK FOR ERROR & RETURN.


9780                             PROC
                        ;
                        ; TOUT -- GRAPHICS DATA OUTPUT
```

```
                      ;
                      ; CALLING SEQUENCE:
                      ;
                      ;       A = ONE GRAPHICS PIXEL
                      ;
                      ;       JSR     TOUT
                      ;
9780  86A1            TOUT    STX     TEMP            ; SAVE REGISTERS.
9782  84A2                    STY     TEMP+1
9784  A216                    LDX     #SPUTC-IOVBAS   ; PUT CHARACTER TO 'S:'.
9786  20F797                  JSR     IOHAND
9789  4C8E94                  JMP     IOERCK          ; CHECK FOR ERROR & RETURN.



978C                          PROC

978C  A920            BUFPNT  LDA     # LOW OPNBUF    ; POINT TO NAME BUFFER FOR OPEN.
978E  9D4403                  STA     ICBAL,X
9791  A905                    LDA     # HIGH OPNBUF
9793  9D4503                  STA     ICBAH,X
9796  60                      RTS

9797                          PROC
                      ;
                      ; PRTSTG -- PRINT TEXT DATA
                      ;
                      ; CALLING SEQUENCE:
                      ;
                      ;       X = OFFSET TO TEXT DATA POINTER.
                      ;
                      ;       JSR     PRTSTG
                      ;
9797  84AA            PRTSTG  STY     TEMP2+3
9799  B580                    LDA     DTAB,X          ; MOVE POINTER.
979B  85A7                    STA     TEMP2
979D  B581                    LDA     DTAB+1,X
979F  85A8                    STA     TEMP2+1
97A1  B583                    LDA     DTAB+3,X        ; ENDING INDEX.
97A3  85A9                    STA     TEMP2+2
97A5  B482                    LDY     DTAB+2,X        ; STARTING INDEX.

97A7  C4A9            :PR010  CPY     TEMP2+2         ; COMPARE START INDEX WITH END INDEX.
97A9  F009 ^97B4              BEQ     :PR080          ; EQUAL -- DONE.

97AB  B1A7                    LDA     (TEMP2),Y       ; GET NEXT CHARACTER.
97AD  C8                      INY
97AE  20F294                  JSR     CHOT            ; PRINT CHARACTER.
97B1  4CA797                  JMP     :PR010

97B4  A4AA            :PR080  LDY     TEMP2+3
97B6  60                      RTS
```

```
97B7                          PROC
                      ;
                      ; IOHAND -- DIRECT I/O TO INTERFACE ROUTINE
                      ;
                      ; CALLING SEQUENCE:
                      ;
                      ;       X = I/O ROUTINE OFFSET TO ADDRESS TABLE ENTRY (SYSTEM)
                      ;
                      ;       JSR     IOHAND
                      ;
                      ;       CLOBBERS Y REGISTER.
                      ;
97B7  A8              IOHAND  TAY                       ; SAVE REGISTER A.
97B8  BD01E4                  LDA     IOVBAS+1,X        ; GET ADDRESS MSB.
97BB  48                      PHA
97BC  BD00E4                  LDA     IOVBAS+0,X        ; GET ADDRESS LSB.
97BF  48                      PHA
97C0  98                      TYA                       ; RESTORE REGISTER A.
97C1  60                      RTS                       ; (JMP) TO HANDLER.
```

```
97C2                            PROC
                        ;
                        ; SFNAME -- GET DEVICE NAME AND STORE IN 'OPNBUF'.
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       'EXEC = 0 FOR SCAN MODE, ELSE EXECUTE.
                        ;       'XXXX' = INPUT LINE INDEX.
                        ;       X = INDEX TO EOL IN 'OPNBUF'.
                        ;
                        ;       JSR     SFNAME
                        ;       BNE     ERROR (A = ERROR CODE).
                        ;
                        ;       'OPNBUF' = DEVICE NAME.
                        ;       Y = INPUT LINE INDEX TO FIELD AFTER DEVICE/FILENAME.
                        ;
97C2  206E81            SFNAME  JSR     ATOM            ; GET DEVICE/FILENAME.
97C5  D00A ^97D1                BNE     :SF090          ; ERROR.

97C7  C920                      CMP     #TEXT           ; TEXT LITERAL?
97C9  F007 ^97D2                BEQ     :SF100          ; YES.

97CB  2918                      AND     #SVAR+USVAR     ; STRING NAME?
97CD  D00C ^97DB                BNE     :SF200          ; YES.

97CF  A902                      LDA     #IMPERR         ; NO -- ERROR.

97D1  60                :SF090  RTS                     ; RETURN WITH CC SET.

                        ; SCAN TEXT LITERAL DATA TO EXTRACT DEVICE/FILENAME.

97D2  20DC96            :SF100  JSR     FNAME           ; NAME TO 'OPNBUF'.
97D5  8C4805                    STY     XXXX            ; SAVE LINE INDEX.
97D8  A900                      LDA     #0              ; SET CC FOR NORMAL EXIT.
97DA  60                        RTS                     ; RETURN WITH CC SET.

                        ; DEVICE/FILENAME IS A STRING VARIABLE VALUE

97DB  A592              :SF200  LDA     EXEC            ; EXECUTE MODE?
97DD  F0F2 ^97D1                BEQ     :SF090          ; NO -- DONE.

97DF  8C4805                    STY     XXXX
97E2  A200                      LDX     #0
97E4  A4C4                      LDY     DP+2

97E6  C4C5              :SF220  CPY     DP+3            ; DONE?
97E8  F00B ^97F5                BEQ     :SF250          ; YES.

97EA  B1C2                      LDA     (DP),Y          ; NO -- MOVE NAME.
97EC  9D2005                    STA     OPNBUF,X
97EF  C8                        INY
97F0  E8                        INX
97F1  E00F                      CPX     #ONSIZE         ; OVERLENGTH NAME?
97F3  D0F1 ^97E6                BNE     :SF220          ; O.K. SO FAR.

97F5  A99B              :SF250  LDA     #EOL
```

```
97F7  9D2005            STA     OPNBUF,X

97FA  A900              LDA     #0                ; SET CC FOR NORMAL EXIT.
97FC  60                RTS                       ; RETURN WITH CC SET.
```

```
97FD                           PROC
                       ;
                       ; SCNDEV -- GET DEVICE NAME AND SETUP FOR 'READ:', 'WRITE:' OR 'CLOSE:'
                       ;
                       ; CALLING SEQUENCE:
                       ;
                       ;        'EXEC' = 0 FOR SCAN MODE, ELSE EXECUTE.
                       ;        Y = INPUT LINE INDEX.
                       ;        A = AUX1 OPEN CODE. (0 = CLOSE, ELSE OPEN).
                       ;
97FD  85E0             SCNDEV  STA     LEND              ; SAVE DEVICE OPEN CODE.
97FF  20C297                   JSR     SFNAME            ; EXTRACT FILENAME.
9802  D034 ^9838               BNE     :SC290            ; ERROR.

9804  A592                     LDA     EXEC              ; EXECUTE MODE?
9806  F030 ^9838               BEQ     :SC290            ; NO -- ALL DONE.

9808  A900                     LDA     #0
980A  85E4                     STA     IOSTAT            ; CLEAR I/O STATUS.
980C  85C0                     STA     NP+2

980E  86C1                     STX     NP+3
9810  A920                     LDA     # LOW OPNBUF
9812  85BE                     STA     NP
9814  A905                     LDA     # HIGH OPNBUF
9816  85BF                     STA     NP+1

9818  20AD9E                   JSR     SETSVL            ; SETUP TO ACCESS STRING VARIABLE LIST.
981B  A920                     LDA     #ATRIO            ; 'I/O' ATTRIBUTE.
981D  8D6605                   STA     ATRTYP
9820  20CE98                   JSR     SFIND             ; SEE IF STRING EXIS:T.
9823  D014 ^9839               BNE     :SC300            ; NO.


9825  A4C4                     LDY     DP+2              ; YES -- GET IOCB INDEX FROM VALUE.
9827  B1C2                     LDA     (DP),Y
9829  48                       PHA
982A  A5E0                     LDA     LEND               ; LOOK AT "OPEN" CODE.
982C  D003 ^9831               BNE     :SC270            ; NORMAL IN OR OUT.

982E  20EC98                   JSR     SDELET            ; 'DONE' -- DELETE NAME.

9831  68               :SC270  PLA
9832  AA                       TAX
9833  AC4805                   LDY     XXXX
9836  A900                     LDA     #0                ; SET CC FOR NORMAL EXIT.

9838  60               :SC290  RTS                       ; RETURN WITH CC SET.

                       ; FIRST ACCESS TO DEVICE, DO IMPLICIT OPEN.

9839  20E798           :SC300  JSR     CHKDEV            ; CHECK FOR VALID DEVICE.

983C  A5E0                     LDA     LEND               ; CHECK "OPEN" CODE.
983E  D006 ^9846               BNE     :SC310            ; NORMAL IN OR OUT.

9840  AC4805                   LDY     XXXX              ; RESTORE INDEX.
```

```
 9843  A902                 LDA    #IMPERR          ; 'DONE' -- CLOSING NON-OPEN FILE.
 9845  60                   RTS

 9846  207098       :SC310  JSR    FNDIOB           ; FIND A FREE IOCB, IF AVAILABLE.
 9849  D024 ^986F           BNE    :SC900           ; NONE AVAILABLE.

 984B  A5ED                 LDA    LEND              ; GET AUX1 OPEN CODE.
 984D  20FA96               JSR    DOPEN            ; OPEN DEVICE.

 9850  86DE                 STX    LS               ; SAVE IOCB # ASSOCIATED WITH DEVICE.
 9852  8EF002               STX    CRSINH           ; INHIBIT CURSOR JUST IN CASE.

 9855  A9DE                 LDA    # LOW LS
 9857  85C2                 STA    DP
 9859  A900                 LDA    # HIGH LS
 985B  85C3                 STA    DP+1
 985D  A900                 LDA    #0
 985F  85C4                 STA    DP+2
 9861  A901                 LDA    #1
 9863  85C5                 STA    DP+3

 9865  200599               JSR    SINSRT           ; INSERT NAMED STRING CONTAINING INFO.
 9868  08                   PHP
 9869  A6DE                 LDX    LS
 986B  AC4805               LDY    XXXX
 986E  28                   PLP

 986F  60           :SC900  RTS                     ; RETURN WITH CC SET.


 9870                       PROC
                    ;
                    ; FNDIOB -- FIND A FREE IOCB
                    ;
                    ; CALLING SEQUENCE:
                    ;
                    ;       JSR    FNDIOB
                    ;       BNE    NO FREE IOCB (A = ERROR CODE)
                    ;
                    ;       X = = IOCB INDEX.
                    ;
 9870  A240         FNDIOB  LDX    #IOCB4           ; START WITH IOCB #4.

 9872  BD4003       :FD010  LDA    ICHID,X          ; TEST FOR CURRENTLY UNUSED.
 9875  C9FF                 CMP    #$FF
 9877  F007 ^9880           BEQ    :FD090           ; FOUND ONE.

 9879  208C98               JSR    NXTIOB           ; BUMP INDEX TO NEXT IOCB.
 987C  D0F4 ^9872           BNE    :FD010           ; MORE TO CHECK.

 987E  A996                 LDA    #FILERR          ; NONE AVAILABLE.

 9880  60           :FD090  RTS                     ; RETURN WITH CC SET.
```

```
9881                          PROC
                      ;
                      ; CLOSEM -- CLOSE IOCBS 3 THROUGH 7 (WHETHER OPEN OR NOT).
                      ;
9881  A230            CLOSEM  LDX     #IOCB3          ; START WITH IOCB #3.

9883  203F97          :CL010  JSR     DCLOSE          ; CLOSE THE IOCB.
9886  208C98                  JSR     NXTIOB          ; BUMP INDEX TO NEXT IOCB.
9889  D0F8 ^9883              BNE     :CL010          ; MORE TO DO.

988B  60                      RTS


988C                          PROC
                      ;
                      ; NXTIOB -- BUMP INDEX TO NEXT IOCB.
                      ;
                      ; CALLING SEQUENCE:
                      ;
                      ;       X = IOCB INDEX
                      ;
                      ;       JSR     NXTIOB
                      ;       BEQ     INDEX PAST IOCB #7
                      ;
                      ;       X = IOCB INDEX TO NEXT IOCB
                      ;
988C  8A              NXTIOB  TXA
988D  18                      CLC
988E  6910                    ADC     #IOCBSZ
9890  AA                      TAX
9891  E080                    CPX     #IOCB7+IOCBSZ

9893                  RDV090
9893                  CKD090
9893  60                      RTS                     ; RETURN WITH CC SET.


9894                          PROC
                      ;
                      ; REMDEV -- REMOVE DEVICE ASSIGNMENTS FROM STRING LIST
                      ;
9894  A252            :RD000  LDX     #MENA-DTAB      ; REMOVE STRING VAR FROM LIST.
9896  A03A                    LDY     #LP-DTAB
9898  20459A                  JSR     CMOVI

989B  203E9B                  JSR     MDEALL

989E  20AD9E          REMDEV  JSR     SETSVL          ; SETUP TO SCAN STRING VARIABLES ...
98A1  A23A                    LDX     #LP-DTAB        ; ... TO REMOVE ALL DEVICE ASSIGNMENTS.

98A3  20139A          :RD010  JSR     SEND            ; END OF LIST?
98A6  F0EB ^9893              BEQ     RDV090          ; YES.

98A8  A23A                    LDX     #LP-DTAB        ; CHECK ATTRIBUTE.
98AA  20869A                  JSR     SATTR
```

```
98AD  C920                    CMP      #ATRIC
98AF  F0E3 ^9894              BEQ      :RD000         ; YES -- REMOVE IT FROM LIST.

98B1  204A9A                  JSR      SNXTI          ; GO TO NEXT ITEM IN LIST.
98B4  4C8396                  JMP      :RD010



98B7                          PROC
                ; CHKDEV -- CHECK FOR VALID DEVICE

98B7  20B896        CHKDEV    JSR      TSTMOD         ; CHECK SCREEN MODE.
98BA  C901                    CMP      #TXSL          ; TEXT, SMALL LETTERS?
98BC  F0D5 ^9893              BEQ      CKD090         ; YES -- NO RESTRICTIONS.

98BE  AD2005                  LDA      OPNBUF         ; CHECK FOR 'E' OR 'S'.
98C1  C945                    CMP      #'E'
98C3  F004 ^98C9              BEQ      :CK010         ; INVALID -- CLOBBERS SCREEN.

98C5  C953                    CMP      #'S'
98C7  D0CA ^9893              BNE      CKD090

98C9  A985          :CK010    LDA      #SCNERR
98CB  4C3A7A                  JMP      PSTOP
```

```
                    ;
                    ; THIS PACKAGE HAS THREE LEVELS OF STRING HANDLING ROUTINES:
                    ;
                    ;       NAMED STRING HANDLING -- SFIND, SDELET & SINSRT
                    ;
                    ;       TEXT DATA HANDLING --  SCOMP
                    ;
                    ;       IMPLEMENTATION UTILITIES -- IFIND, SEND, PSETUP, PMOVE, ICOMP,
                    ;                                   ILENG,  SNXTI, IMATCH & IALLOC
                    ;
```

```
                         ;
                         ; NAMED STRING HANDLING
                         ;
                         ; THESE ROUTINES USE THE FOLLOWING VARIABLES:
                         ;
                         ;       NP = POINTER TO STRING NAME.
                         ;       DP = POINTER TO STRING DATA PORTION.
                         ;       LP = POINTER TO START OF LIST OF NAMED STRINGS (S1L OR S2L).
                         ;


98CE                             PROC
                         ;
                         ; SFIND -- FIND NAMED STRING IN LIST
                         ;
                         ; CALLING SEQUENCE:
                         ;
                         ;       LP POINTS TO START OF LIST OF NAMED STRINGS
                         ;       NP POINTS TO NAME TO FIND IN LIST
                         ;       ATRTYP  SET
                         ;
                         ;       JSR     SFIND
                         ;       BNE     NAME NOT IN LIST OR NAME IS NULL
                         ;
                         ;       DP POINTS TO DATA PORTION OF NAMED STRING FOUND IN LIST
                         ;
98CE  205399             SFIND   JSR     IFIND            ; FIND NAME IN LIST.
98D1  D018 ^98EB                 BNE     :SF080           ; NOT FOUND.

98D3  A242                       LDX     #DP-DTAB         ; SET 'DP' TO POINT TO DATA PORTION.
98D5  A04E                       LDY     #PP-DTAB
98D7  20459A                     JSR     DMOVI

98DA  A5D1                       LDA     PP+3             ; SKIP OVER NAME PORTION.
98DC  20089D                     JSR     DADDP

98DF  A901                       LDA     #1               ; SET START INDEX.
98E1  85C4                       STA     DP+2

98E3  A000                       LDY     #0               ; SET END INDEX.
98E5  18                         CLC
98E6  71C2                       ADC     (DP),Y
98E8  85C5                       STA     DP+3

98EA  98                         TYA                      ; SET CC FOR EXIT.

98EB  60                 :SF080  RTS                      ; RETURN WITH CC SET.


98EC                             PROC
                         ;
                         ; SDELET -- DELETE NAMED STRING FROM LIST
                         ;
                         ; CALLING SEQUENCE:
                         ;
```

```
                        ;       NP POINTS TO STRING NAME
                        ;       LP POINTS TO START OF LIST OF NAMED STRNGS
                        ;       ATRTYP  SET
                        ;
                        ;       JSR     SDELFT
                        ;       BNE     NAMED STRING NOT FOUND OR NAME IS NULL
                        ;
98EC  205899            SDELET  JSR     IFIND           ; FIND STRING IN LIST.
98EF  D013 ^9904                BNE     :SD090          ; NAMED STRING NOT FOUND.

                        ; *** EXTERNAL ENTRY POINT ***

98F1  A252              SDEL2   LDX     #MEMA-DTAB      ; MEMA = PP (FOR DEALLOCATE CALL).
98F3  A04E                      LDY     #PP-DTAB
98F5  20459A                    JSR     DMOVI

98F8  203E96                    JSR     MDEALL          ; DELETE STRING.

98FB  A24E                      LDX     #PP-DTAB        ; PP = MEMA.
98FD  A052                      LDY     #MEMA-DTAB
98FF  20459A                    JSR     DMOVI

9902  A900                      LDA     #0              ; SET CC FOR NORMAL EXIT.

9904  60                :SD090  RTS                     ; RETURN WITH CC SET.


9905                            PROC
                        ;
                        ; SINSRT -- NAMED STRING INSERT
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       NP POINTS TO STRING NAME
                        ;       DP POINTS TO STRING DATA PORTION
                        ;       LP POINTS TO START OF NAMED STRING LIST
                        ;       ATRTYP SET
                        ;       TKNTYP, TKNOFF SET IF NUMBERED STATEMENT.
                        ;
                        ;       JSR     SINSRT
                        ;       BNE     NAME IS NULL, OR NO ROOM FOR STRING IN LIST
                        ;
                        ;       ATRTYP  STORED IF 'VARIABLE'
                        ;       TKNTYP, TKNOFF  STORED IF NUMBERED STATEMENT.
                        ;
9905  205899            SINSRT  JSR     IFIND           ; IS NAME ALREADY IN LIST?
9908  D003 ^990D                BNE     :SI020          ; NO.

990A  20F198                    JSR     SDEL2           ; YES -- DELETE OLD OCCURRENCE.

990D  A900              :SI020  LDA     #0              ; CALCULATE ALLOCATION SIZE.
990F  85D4                      STA     MEMB
9911  85D5                      STA     MEMB+1

9913  A254                      LDX     #MEMB-DTAB      ; STRING SIZE = NAME SIZE ...
9915  38                        SEC
```

```
9916  A5C1                 LDA    NP+3
9918  E5C0                 SBC    NP+2
991A  20089D               JSR    DADDP

991D  38                   SEC                    ; ... + DATA PORTION SIZE ...
991E  A5C5                 LDA    DP+3
9920  E5C4                 SBC    DP+2
9922  20089D               JSR    DADDP

9925  A906                 LDA    #6              ; ... + 6 BYTES OF OVERHEAD.
                                                  ; 2 = BLOCK SIZE; 1 = NAME SIZE,
                                                  ; 1 = DATA SIZE; 2 = 'EXTRA' AT END.
9927  20089D               JSR    DADDP

992A  A252                 LDX    #MEMA-DTAB      ; ALLOCATE ADDRESS FROM 'IFIND' CALL IN PP.
992C  A04E                 LDY    #PP-DTAB
992E  20459A               JSR    DMOVI

9931  20C19A               JSR    MALLOC          ; ALLOCATE SPACE IN LIST.
9934  D01E ^9954           BNE    :SI090          ; NOT ENOUGH ROOM.

9936  A23E                 LDX    #NP-DTAB        ; MOVE NAME TO NEW STRING ...
9938  A002                 LDY    #2              ; ... STARTING AFTER ALLOCATION SIZE.
993A  20509A               JSR    SMOVI

993D  A242                 LDX    #DP-DTAB        ; NOW MOVE DATA PORTION.
993F  20509A               JSR    SMOVI

                   ; 'MEMA' = ADDRESS OF 'ATTRIBUTE' DESTINATION.
                   ; Y = 0.

9942  AD6605               LDA    ATRTYP
9945  D009 ^9950           BNE    :SI060          ; 'VARIABLE' ATTRIBUTE.

9947  AD6805               LDA    TKNTYP          ; TOKENIZE LINE.
994A  9102                 STA    (MEMA),Y
994C  C8                   INY
994D  AD6A05               LDA    TKNOFF          ; OFFSET.

9950  91D2         :SI060  STA    (MEMA),Y
9952  A900                 LDA    #0              ; SET CC FOR NORMAL EXIT.

9954  60           :SI090  RTS                    ; RETURN WITH CC SET.
```

```
                      ;
                      ; TEXT DATA UTILITIES
                      ;
                      ; THESE ROUTINES USE THE FOLLOWING VARIABLES:
                      ;
                      ;       DP = POINTER TO TEXT DATA
                      ;       MP = POINTER TO TEXT PATTERN DATA
                      ;       AP1 = AUXILLIARY POINTER TO TEXT SUB-STRING
                      ;       AP2 = AUXILLIARY POINTER TO TEXT SUB-STRING




9955                          PROC
                      ;
                      ; SCOMP -- COMPARE TWO TEXT STRINGS
                      ;
                      ; CALLING SEQUENCE:
                      ;
                      ;       DP POINTS TO DATA TEXT
                      ;       MP POINTS TO DATA TEXT
                      ;
                      ;       JSR     SCOMP
                      ;       BEQ     DATA TEXTS ARE IDENTICAL
                      ;       BCS     DP TEXT >= MP TEXT
                      ;       BCC     DP TEXT < MP TEXT
                      ;
                      ; NOTE: THE COMPARISON IS BASED UPON THE STANDARD ATASCII COLLATION
                      ;       SEQUENCE; WHEN ONE TEXT IS A SUBSET OF THE FIRST PART OF THE
                      ;       OTHER TEXT, THE SHORTER ONE IS CONSIDERED TO BE < THE LONGER ONE.
                      ;
9955  2029A           SCOMP   JSR     PSETUP          ; DP TO SP, MP TO PP.

9958  4CA999                  JMP     ICOMP           ; COMPARE TEXT & RETURN WITH CC SET.
```

```
                    ;
                    ; GENERAL STRING IMPLEMENTATION UTILITIES
                    ;
                    ; THESE ROUTINES USE THE FOLLOWING VARIABLES:
                    ;
                    ;       SP = SOURCE TEXT POINTER
                    ;       PP = PATTERN TEXT POINTER


995B                        PROC
                    ;
                    ; IFIND -- FIND NAMED STRING IN LIST
                    ;
                    ; CALLING SEQUENCE:
                    ;
                    ;       NP POINTS TO DESIRED NAME
                    ;       LP POINTS TO START OF NAMED STRING LIST
                    ;       ATRTYP  ATTRIBUTE
                    ;
                    ;       JSR     IFIND
                    ;       BNE     NOT FOUND  (PP POINTS TO SUCCESSOR)
                    ;
                    ;       PP POINTS TO NAMED STRING IN LIST
                    ;
                    ;       IF NOT FOUND, THE SUCCESSOR IS CHOSEN SO THAT:
                    ;
                    ;       STATEMENTS ARE KEPT IN LINE # ORDER.
                    ;       VARIABLES ARE APPENDED TO THE END OF THE LIST.
                    ;
995B  A5C0          IFIND   LDA     NP+2            ; NAME NULL?
995D  C5C1                  CMP     NP+3
995F  F045 ^99A6            BEQ     :IF080          ; YES -- DONE.

9961  A24A                  LDX     #SP-DTAB        ; SP = NP.
9963  A03E                  LDY     #NP-DTAB
9965  20389A                JSR     PMOVE

9968  A24E                  LDX     #PP-DTAB        ; PP = LP.
996A  A03A                  LDY     #LP-DTAB
996C  20389A                JSR     PMOVE

996F                :IF020
996F  A24E                  LDX     #PP-DTAB
9971  20139A                JSR     SEND            ; END OF LIST?
9974  F030 ^99A6            BEQ     :IF080          ; YES -- DONE.

9976  A903                  LDA     #3              ; NO -- SETUP START INDEX ...
9978  85D0                  STA     PP+2
997A  18                    CLC                     ; ... & END INDEX (TO NAME).
997B  A002                  LDY     #2
997D  71CE                  ADC     (PP),Y
997F  85D1                  STA     PP+3

9981  20A999                JSR     ICOMP           ; NAME COMPARISON.
9984  F009 ^998F            BEQ     :IF030          ; A MATCH.
9986  B016 ^999E            BCS     :IF040          ; NOT THERE YET (IF LINE).
```

```
                                                   ; NOT A MATCH (IF VARIABLE).
9988  AD6605             LDA     ATRTYP            ; LINE INSERTION?
998B  D011 ^999E         BNE     :IF040            ; NO -- SEARCH TO END.
998D  9017 ^99A6         BCC     :IF080            ; YES -- PAST CORRECT SPOT (BRA).

                  ; CHECK IF ATTRIBUTE MATCHES.

998F  AD6605    :IF030   LDA     ATRTYP            ; ATTRIBUTE TO MATCH.
9992  F014 ^99A8         BEQ     :IF090            ; 'LINE' -- FOUND IT!

9994  A24E               LDX     #PP-DTAB          ; CHECK ATTRIBUTE.
9996  20B69A             JSR     SATTR
9999  CD6605             CMP     ATRTYP
999C  F00A ^99A8         BEQ     :IF090            ; ATTRIBUTE MATCHED!

999E  A24E      :IF040   LDX     #PP-DTAB          ; SKIP TO NEXT LIST ENTRY.
99A0  20AA9A             JSR     SNXTI

99A3  4C6F99             JMP     :IF020            ; TRY AGAIN.

99A6  A9FF      :IF080   LDA     #$FF              ; SET CC FOR EXIT (NOT FOUND).

99A8  60        :IF090   RTS                       ; RETURN WITH CC SET.


99A9                     PROC
                ;
                ; ICOMP -- COMPARE TEXT DATA
                ;
                ; CALLING SEQUENCE:
                ;
                ;        SP POINTS TO DATA TEXT
                ;        PP POINTS TO DATA TEXT
                ;
                ;        JSR     ICOMP
                ;        BEQ     DATA TEXTS ARE IDENTICAL
                ;        BCS     SP DATA >= PP DATA
                ;        BCC     SP DATA < PP DATA
                ;
99A9  201F9A    ICOMP    JSR     ILENG             ; SEE IF EQUAL LENGTHS.
99AC  F03C ^99EA         BEQ     IMATCH            ; YES -- COMPARE & RETURN.

99AE  B01D ^99CD         BCS     :IC050            ; PP DATA SHORTER THAN SP DATA.

99B0  A5D1               LDA     PP+3              ; SAVE STARTING VALUE.
99B2  85A7               STA     TEMP2
                                                   ; (CLEAR BORROW).
99B4  38                 SEC                       ; ADJUST PP DATA LENGTH FOR COMPARISON.
99B5  A5CD               LDA     SP+3
99B7  E5CC               SBC     SP+2
99B9  18                 CLC
99BA  65D0               ADC     PP+2
99BC  85D1               STA     PP+3

99BE  20EA99             JSR     IMATCH            ; NOW COMPARE.
99C1  08                 PHP
99C2  A5A7               LDA     TEMP2             ; RESTORE ALTERED PARAMETER.
```

```
 99C4  85D1                STA    PP+3
 99C6  28                  PLP
 99C7  D020 ^99E9          BNE    :IC090          ; NOT EQUAL -- CC SET FOR EXIT.

 99C9  A9FF                LDA    #$FF            ; SET CC FOR EXIT.
 99CB  18                  CLC
 99CC  60                  RTS                    ; RETURN WITH CC SET.

 99CD  A5CD         :IC050 LDA    SP+3            ; SAVE STARTING VALUE.
 99CF  85A7                STA    TEMP2
 99D1  18                  CLC
 99D2  A5CC                LDA    SP+2            ; ADJUST SP LENGTH FOR COMPARISON.
 99D4  65D1                ADC    PP+3
 99D6  38                  SEC
 99D7  E5D0                SBC    PP+2
 99D9  85CD                STA    SP+3

 99DB  20EA99              JSR    IMATCH          ; NOW COMPARE.
 99DE  08                  PHP
 99DF  A5A7                LDA    TEMP2           ; RESTORE ALTERED PARAMETER.
 99E1  85CD                STA    SP+3
 99E3  28                  PLP
 99E4  D003 ^99E9          BNE    :IC090          ; NOT EQUAL -- CC SET FOR EXIT.

 99E6  A9FF                LDA    #$FF            ; SET CC FOR EXIT.
 99E8  38                  SEC

 99E9  60           :IC090 RTS                    ; RETURN WITH CC SET.


 99EA                      PROC
                    ;
                    ; IMATCH -- MATCH TWO TEXT DATA STRINGS
                    ;
                    ; CALLING SEQUENCE:
                    ;
                    ;        SP = SOURCE DATA TEXT (SOURCE DATA MUST BE LONGER THAN PATTERN)
                    ;        PP = PATTERN DATA TEXT
                    ;
                    ;        JSR     IMATCH
                    ;        BEQ     PATTERN IS CONTAINED WITHIN SOURCE
                    ;        BCS     SOURCE COLLATES >= PATTERN
                    ;        BCC     SOURCE COLLATES < PATTERN
                    ;
 99EA  A5CC         IMATCH LDA    SP+2            ; SAVE STARTING INDICES.
 99EC  85A1                STA    TEMP
 99EE  A5D0                LDA    PP+2
 99F0  85A2                STA    TEMP+1

 99F2  A4D0         :IM010 LDY    PP+2            ; SEE IF ALL OF PATTERN HAS MATCHED.
 99F4  C4D1                CPY    PP+3
 99F6  F010 ^9A08          BEQ    :IM090          ; YES -- ALL DONE.

 99F8  A4CC                LDY    SP+2            ; NO -- COMPARE ANOTHER BYTE.
 99FA  B1CA                LDA    (SP),Y
 99FC  E6CC                INC    SP+2
```

```
99FC  E6CC                    INC      SP+2
```

```
99FE  A4D0                    LDY      PP+2
9A00  D1CE                    CMP      (PP),Y
9A02  D004 ^9A08              BNE      :IM090          ; NO COMPARE -- CC SET FOR EXIT.

9A04  E6D0                    INC      PP+2
9A06  B0EA ^99F2              BCS      :IM010          ; (BRA).

9A08  08              :IM090  PHP                      ; SAVE CC.
9A09  A5A1                    LDA      TEMP            ; RESTORE STARTING INDICES.
9A0B  85CC                    STA      SP+2
9A0D  A5A2                    LDA      TEMP+1
9A0F  85D0                    STA      PP+2
9A11  28                      PLP                      ; RESTORE CC.

9A12                  SEN090
9A12  60                      RTS                      ; RETURN WITH CC SET.


9A13                          PROC
                      ;
                      ; SEND -- CHECK FOR END OF STRING LIST
                      ;
                      ; CALLING SEQUENCE:
                      ;
                      ;       X = DTAB OFFSET TO LIST POINTER
                      ;
                      ;       JSR     SEND
                      ;       BEQ     END OF LIST REACHED
                      ;
                      ;       Y IS ALTERED
                      ;
9A13  A030            SEND    LDY      #S1E-DTAB       ; SEE IF END OF REGION #1.
9A15  20159C                  JSR      DCMPI
9A18  F0F8 ^9A12              BEQ      SEN090          ; YES.

9A1A  A034                    LDY      #S2E-DTAB       ; SEE IF END OF REGION #2 ...
9A1C  4C159C                  JMP      DCMPI           ; ... & RETURN WITH CC SET.


9A1F                          PROC
                      ;
                      ; ILENG -- COMPARE LENGTHS OF SOURCE TEXT AND PATTERN TEXT
                      ;
                      ; CALLING SEQUENCE:
                      ;
                      ;       SP POINTS TO SOURCE DATA TEXT
                      ;       PP POINTS TO PATTERN DATA TEXT
                      ;
                      ;       JSR     ILENG
                      ;       BEQ     DATA TEXTS ARE EQUAL LENGTH
                      ;       BCS     SOURCE TEXT >= PATTERN TEXT
                      ;       BCC     SOURCE TEXT < PATTERN TEXT
                      ;
9A1F  A5D1            ILENG   LDA      PP+3
9A21  38                      SEC
```

```
9A22  E5D0                SBC     PP+2
9A24  85A1                STA     TEMP

9A26  A5CC                LDA     SP+3
9A28  E5CC                SBC     SP+2

9A2A  E5A1                SBC     TEMP             ; CC = SP LENGTH - PP LENGTH.
9A2C  60                  RTS


9A2D                      PROC
                ;
                ; PSETUP -- MOVE POINTERS (DP TO SP, MP TO PP)
                ;
                ; CALLING SEQUENCE:
                ;
                ;         JSR     PSETUP
                ;
                ;         SP = DP
                ;         PP = MP
                ;
9A2D  A24A      PSETUP    LDX     #SP-DTAB         ; SP = DP.
9A2F  A042                LDY     #DP-DTAB
9A31  203B9A              JSR     PMOVE

9A34  A24E                LDX     #PP-DTAB         ; PP = MP.
9A36  A046                LDY     #MP-DTAB
9A38  4C3B9A              JMP     PMOVE            ; AND RETURN.


9A3B                      PROC
                ;
                ; PMOVE -- MOVE STRING/DATA TEXT POINTERS
                ;
                ; CALLING SEQUENCE:
                ;
                ;         X = DTAB OFFSET
                ;         Y = DTAB OFFSET
                ;
                ;         JSR     PMOVE
                ;
                ;         DTAB(X) = DTAB(Y) (4 BYTE MOVE)
                ;
9A3B  B98200    PMOVE     LDA     DTAB+2,Y
9A3E  9582                STA     DTAB+2,X

9A40  B98300              LDA     DTAB+3,Y
9A43  9583                STA     DTAB+3,X

                ; *** EXTERNAL ENTRY POINT ***

9A45  B98000    DMOVI     LDA     DTAB,Y
9A48  9580                STA     DTAB,X

9A4A  B98100              LDA     DTAB+1,Y
```

```
9A4D  9581              STA     DTAB+1,X

9A4F  60                RTS


      = 0000            IF      FALSE
    -                   PROC
                ;
                ; IALLOC -- ALLOCATE MEMORY
                ;
                ; CALLING SEQUENCE:
                ;
                ;       A = # OF BYTES TO ALLOCATE
                ;
                ;       JSR     IALLOC
                ;       BNE     NOT ENOUGH ROOM
                ;
                ;       DP POINTS TO NEW ALLOCATION + 2 (START OF STRING)
                ;
    -           IALLOC  STA     MEMB            ; SETUP MEMB = # BYTES ...
    -                   LDA     #0
    -                   STA     MEMB+1

    -                   LDA     #3              ; ... + 3.
    -                   LDX     #MEMB-DTAB
    -                   JSR     DADDS

    -                   LDX     #MEMA-DTAB      ; SETUP MEMA = ALLOCATION ADDRESS.
    -                   LDY     #S2L-DTAB
    -                   JSR     DMOVI

    -                   JSR     MALLOC          ; ALLOCATE MEMORY.
    -                   BNE     :IA090          ; NOT ENOUGH ROOM.

    -                   LDX     #DP-DTAB        ; DP = ADDRESS OF STRING STORAGE AREA.
    -                   LDY     #MEMA-DTAB
    -                   JSR     DMOVI

    -                   LDA     #2
    -                   JSR     DADDS

    -                   LDA     #1              ; SET STARTING & ENDING INDICES.
    -                   STA     DP+2
    -                   STA     DP+3
    -                   LDA     #0              ; SET CC FOR EXIT.

    -           :IA090  RTS                     ; RETURN WITH CC SET.
                        ENDIF


9A50                    PROC
                ;
                ; SMOVI -- MOVE TEXT DATA TO MEMA (FORMING STRING)
                ;
                ; CALLING SEQUENCE:
```

```
                          ;
                          ;       X = DTAB INDEX TO STRING POINTER
                          ;       Y = MEMA OFFSET TO START STORING
                          ;
                          ;       JSR      SMOVI
                          ;
                          ;       MEMA = LAST LOCATION STORED INTO + 1
                          ;       Y = 0
                          ;
9A50  B580          SMOVI  LDA     DTAB,X           ; MOVE SOURCE POINTER TO TEMP.
9A52  85A1                 STA     TEMP
9A54  B581                 LDA     DTAB+1,X
9A56  85A2                 STA     TEMP+1
9A58  B582                 LDA     DTAB+2,X
9A5A  85A3                 STA     TEMP+2
9A5C  B583                 LDA     DTAB+3,X
9A5E  85A4                 STA     TEMP+3

9A60  38                   SEC                      ; CALCULATE STRING LENGTH ...
9A61  A5A4                 LDA     TEMP+3
9A63  E5A3                 SBC     TEMP+2

9A65  9102          :SM010 STA     (MEMA),Y         ; ... & STORE IN TARGET AREA.
9A67  C8                   INY
9A68  84A5                 STY     TEMP+4           ; SAVE INDEX.

9A6A  A4A3                 LDY     TEMP+2           ; DONE?
9A6C  C4A4                 CPY     TEMP+3
9A6E  F008 ^9A78           BEQ     :SM090           ; YES.

9A70  B1A1                 LDA     (TEMP),Y         ; NO -- MOVE A BYTE.
9A72  E6A3                 INC     TEMP+2

9A74  A4A5                 LDY     TEMP+4           ; GET TARGET INDEX.
9A76  D0ED ^9A65           BNE     :SM010           ; (BRA).

9A78  A900          :SM090 LDA     #0               ; PREPARE FOR D.P. ADDITION.
9A7A  85A6                 STA     TEMP+5
9A7C  A252                 LDX     #MEMA-DTAB       ; PREPARE TO BUMP MEMA.
9A7E  A025                 LDY     #TEMP+4-DTAB
9A80  20329C               JSR     DADDI
9A83  A000                 LDY     #0               ; AS PROMISED.
9A85  60                   RTS


9A86                       PROC
                          ;
                          ; SATTR -- POINT TO ATTRIBUTE BYTE
                          ;
                          ; CALLING SEQUENCE:
                          ;
                          ;       X     = DATA OFFSET TO STRING POINTER
                          ;
                          ;       JSR SATTR
                          ;
                          ;       A     = ATTRIBUTE VALUE
```

```
                        ;       TEMP = ADDRESS OF ATTRIBUTE BYTE
                        ;       Y    = 0
                        ;
9A86  B581              SATTR   LDA     DTAB+1,X        ; MOVE POINTER TO TEMP.
9A88  85A2                      STA     TEMP+1
9A8A  B580                      LDA     DTAB,X
9A8C  85A1                      STA     TEMP

9A8E  A000                      LDY     #0              ; ADDRESS ...
9A90  18                        CLC                     ; ...+ LENGTH.
9A91  71A1                      ADC     (TEMP),Y
9A93  48                        PHA                     ; LSB

9A94  C8                        INY
9A95  A5A2                      LDA     TEMP+1
9A97  71A1                      ADC     (TEMP),Y
9A99  85A2                      STA     TEMP+1          ; MSB

9A9B  68                        PLA
9A9C  38                        SEC                     ; ... - 2.
9A9D  E902                      SBC     #2
9A9F  85A1                      STA     TEMP
9AA1  B002 ^9AA5                BCS     :SA010
9AA3  C6A2                      DEC     TEMP+1          ; (BORROW).

9AA5  A000              :SA010  LDY     #0
9AA7  B1A1                      LDA     (TEMP),Y        ; AS ADVERTISED.
9AA9  60                        RTS


9AAA                            PROC
                        ;
                        ; SNXTI -- POINT TO NEXT STRING IN LIST
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       X = DATA OFFSET TO STRING LIST POINTER
                        ;
                        ;       JSR     SNXTI
                        ;
                        ;       DTAB(X) = POINTER TO NEXT STRING IN LIST
                        ;
9AAA  B581              SNXTI   LDA     DTAB+1,X        ; MOVE STRING POINTER TO TEMP.
9AAC  85A2                      STA     TEMP+1
9AAE  B580                      LDA     DTAB,X
9AB0  85A1                      STA     TEMP

9AB2  A000                      LDY     #0              ; ADD ADDRESS TO ...
9AB4  18                        CLC
9AB5  71A1                      ADC     (TEMP),Y        ; ... ALLOCATION LENGTH ...
9AB7  9580                      STA     DTAB,X          ; ... TO GET NEXT ADDRESS.

9AB9  C8                        INY
9ABA  A5A2                      LDA     TEMP+1
9ABC  71A1                      ADC     (TEMP),Y
9ABE  9581                      STA     DTAB+1,X
```

```
9AE0  60                    RTS
```

```
                        ;
                        ; MEMORY MANAGEMENT PACKAGE
                        ;
                        ; AVAILABLE MEMORY IS DIVIDED INTO TWO REGIONS WHICH GROW TOWARD EACH
                        ; OTHER; THE REGIONS ARE DEFINED BY FOUR POINTER VARIABLES:
                        ;
                        ;       'S1L' POINTS TO BOTTOM OF REGION #1
                        ;       'S1H' POINTS TO FIRST UNUSED LOCATION ABOVE REGION #1
                        ;       'S2L' POINTS TO BOTTOM OF REGION #2
                        ;       'S2H' POINTS TO FIRST UNUSED LOCATION ABOVE REGION #2
                        ;
                        ; THREE ROUTINES ARE PROVIDED TO ALLOCATE AND DEALLOCATE MEMORY:
                        ;
                        ;       'MALLOC' IS USED TO ALLOCATE MEMORY
                        ;       'MDEALL' IS USED TO DEALLOCATE MEMORY
                        ;
                        ; THE TWO REGIONS ARE MAINTAINED AS TWO COMPRESSED STACKS; ALLOCATION
                        ; AND DEALLOCATION INVOLVES THE MOVEMENT OF DATA TO CREATE AND
                        ; ELIMINATE HOLES IN THE STACKS.
                        ;



9AC1                            PROC
                        ;
                        ; MALLOC -- MEMORY ALLOCATE
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       'MEMA' CONTAINS THE ADDRESS OF THE START OF ALLOCATION
                        ;          REGION #1: DATA AT START ADDRESS AND ABOVE ARE MOVED UP.
                        ;          REGION #2: DATA BELOW START ADDRESS ARE MOVED DOWN.
                        ;       'MEMB' CONTAINS THE NUMBER OF BYTES TO ALLOCATE
                        ;
                        ;       JSR     MALLOC
                        ;       BNE     NOT ENOUGH MEMORY TO SATISFY ALLOCATION
                        ;
                        ;       'MEMA' CONTAINS LOWEST ADDRESS IN THE ALLOCATED BLOCK
                        ;       FIRST TWO BYTES OF ALLOCATED BLOCK = BLOCK SIZE
                        ;
9AC1  A030              MALLOC  LDY     #S1H-DTAB       ; ACC = S1H ...
9AC3  20A29D                    JSP     DLOADA

9AC6  A054                      LDY     #MEMB-DTAB      ; ... + MEMB.
9AC8  20AC9D                    JSR     DADDA

9ACB  A032                      LDY     #S2L-DTAB       ; COMPARE ACC WITH S2L.
9ACD  20B69D                    JSR     DCMPA
9AD0  B069 ^9B3B                BCS     :MA300          ; NOT ENOUGH ROOM.

9AD2  A252                      LDX     #MEMA-DTAB      ; SEE IF ALLOCATION IN REGION #1 OR #2.
9AD4  A032                      LDY     #S2L-DTAB
9AD6  20159C                    JSR     DCMPI
9AD9  B028 ^9B03                BCS     :MA100          ; REGION #2.

                        ; ALLOCATE FROM REGION #1
```

```
9ADB  A256              LDX    #MSP-DTAB       ; MSP = MEMA.
9ADD  A052              LDY    #MEMA-DTAB
9ADF  20459A            JSR    DMOVI

9AE2  A258              LDX    #MDP-DTAB       ; MDP = MEMA ...
9AE4  20459A            JSR    DMOVI

9AE7  A054              LDY    #MEMB-DTAB      ; ... + MEMB.
9AE9  20329C            JSR    DADDI

9AEC  A25A              LDX    #MBC-DTAB       ; MBC = S1H ...
9AEE  A030              LDY    #S1H-DTAB
9AF0  20459A            JSR    DMOVI

9AF3  A052              LDY    #MEMA-DTAB      ; ... - MEMA.
9AF5  20429C            JSR    DSUBI

9AF8  A230              LDX    #S1H-DTAB       ; S1H = ACC (= S1H + MEMB).
9AFA  20A79D            JSR    DSTORA

9AFD  20CA9B            JSR    MOVDA           ; MOVE DATA UPWARD.

9B00  4C2E9B            JMP    :MA200

                  ; ALLOCATE IN REGION #2

9B03  A256        :MA100  LDX    #MSP-DTAB       ; MSP = S2L.
9B05  A032              LDY    #S2L-DTAB
9B07  20459A            JSR    DMOVI

9B0A  A25A              LDX    #MBC-DTAB       ; MBC = MEMA ...
9B0C  A052              LDY    #MEMA-DTAB
9B0E  20459A            JSR    DMOVI

9B11  A032              LDY    #S2L-DTAB       ; ... - S2L.
9B13  20429C            JSR    DSUBI

9B16  A232              LDX    #S2L-DTAB       ; S2L = S2L - MEMB.
9B18  A054              LDY    #MEMB-DTAB
9B1A  20429C            JSR    DSUBI

9B1D  A258              LDX    #MDP-DTAB       ; MDP = S2L (NEW VALUE).
9B1F  A032              LDY    #S2L-DTAB
9B21  20459A            JSR    DMOVI

9B24  A252              LDX    #MEMA-DTAB      ; MEMA = MEMA - MEMB.
9B26  A054              LDY    #MEMB-DTAB
9B28  20429C            JSR    DSUBI

9B2B  20A69B            JSR    MOVIA           ; MOVE DATA DOWNWARD.

                  ; COMMON CODE

9B2E  A000        :MA200  LDY    #0              ; MOVE BLOCK SIZE TO BLOCK.
9B30  A5D4              LDA    MEMB
9B32  91D2              STA    (MEMA),Y
9B34  C8                INY
```

```
9835  A5D5                 LDA     MEMB+1
9837  91D2                 STA     (MEMA),Y

9839  88                   DEY                        ; SET CC FOR NORMAL EXIT.
983A  60                   RTS

983B  A989         :MA300  LDA     #INSERR            ; SET CC FOR ERROR EXIT.
983D  60                   RTS


983E                       PROC
                   ;
                   ; MDEALL -- MEMORY DEALLOCATE
                   ;
                   ; CALLING SEQUENCE:
                   ;
                   ;       'MEMA' = ADDRESS OF BLOCK TO DEALLOCATE
                   ;       FIRST 2 BYTES OF BLOCK = SIZE OF BLOCK
                   ;
                   ;       JSR     MDEALL
                   ;
                   ;       'MEMA' = ADDRESS OF BLOCK FOLLOWING DEALLOCATED BLOCK (AFTER DEALL)
                   ;
983E  A000         MDEALL  LDY     #0                 ; GET SIZE OF BLOCK TO MEMB.
9840  B1D2                 LDA     (MEMA),Y
9842  85D4                 STA     MEMB
9844  C8                   INY
9845  B1D2                 LDA     (MEMA),Y
9847  85D5                 STA     MEMB+1

9849  A252                 LDX     #MEMA-DTAB         ; SEE IF IN REGION #1 OR #2.
984B  A032                 LDY     #S2L-DTAB
984D  20159C               JSR     DCMPI
9850  B029 ^987B           BCS     :MD100             ; REGION #2.

                   ; DEALLOCATE FROM REGION #1.

9852  A256                 LDX     #MSP-DTAB          ; MSP = MEMA ...
9854  A052                 LDY     #MEMA-DTAB
9856  20459A               JSR     DMOVI

9859  A054                 LDY     #MEMB-DTAB         ; ... + MEMB.
985B  20329C               JSR     DADDI

985E  A25A                 LDX     #MBC-DTAB          ; MBC = S1H ...
9860  A030                 LDY     #S1H-DTAB
9862  20459A               JSR     DMOVI

9865  A056                 LDY     #MSP-DTAB          ; ... - MSP.
9867  20429C               JSR     DSUBI

986A  A230                 LDX     #S1H-DTAB          ; S1H = S1H - MEMB.
986C  A054                 LDY     #MEMB-DTAB
986E  20429C               JSR     DSUBI

9871  A258                 LDX     #MDP-DTAB          ; MDP = MEMA.
```

```
9873  A052                    LDY     #MEMA-DTAB
9875  20459A                  JSR     DMOVI

9878  4CA59B                  JMP     MOVIA               ; MOVE DATA DOWNWARD & RETURN.

                    ; DEALLOCATE MEMORY IN REGION #2

987B  A256          :MD100    LDX     #MSP-DTAB           ; MSP = S2L.
987D  A032                    LDY     #S2L-DTAB
987F  20459A                  JSR     DMOVI

9882  A25A                    LDX     #MBC-DTAB           ; MBC = MEMA ...
9884  A052                    LDY     #MEMA-DTAB
9886  20459A                  JSR     DMOVI

9889  A032                    LDY     #S2L-DTAB           ; ... - S2L.
988B  20429C                  JSR     DSUBI

988E  A232                    LDX     #S2L-DTAB           ; S2L = S2L + MEMB.
9890  A054                    LDY     #MEMB-DTAB
9892  20329C                  JSR     DADDI

9895  A258                    LDX     #MDP-DTAB           ; MDP = S2L (NEW VALUE).
9897  A032                    LDY     #S2L-DTAB
9899  20459A                  JSR     DMOVI

989C  A252                    LDX     #MEMA-DTAB          ; MEMA = MEMA + MEMB.
989E  A054                    LDY     #MEMB-DTAB
98A0  20329C                  JSR     DADDI

98A3  4CCA9B                  JMP     MOVDA               ; MOVE DATA UPWARD & RETURN.
```

```
                    ;
                    ; MOVE UTILITIES FOR MEMORY MANAGEMENT
                    ;
                    ; MOVE BLOCKS OF DATA WITH EITHER INCREASING OR DECREASING ADDRESS
                    ;
                    ; THREE VARIABLES CONTROL THE MOVE ROUTINES:
                    ;
                    ;       'MSP' CONTAINS POINTER TO SOURCE DATA LOCATION
                    ;       'MDP' CONTAINS POINTER TO DESTINATION DATA LOCATION
                    ;       'MBC' CONTAINS THE NUMBER OF BYTES TO MOVE
                    ;


9BA6                        PROC
                    ;
                    ; MOVIA -- MOVE DATA BLOCK WITH INCREASING ADDRESS
                    ;
                    ; CALLING SEQUENCE:
                    ;
                    ;       'MSP', 'MDP' & 'MBC' SETUP
                    ;
                    ;       JSR     MOVIA
                    ;
9BA6  A5DA          MOVIA   LDA     MBC             ; SEE IF BYTE COUNT = ZERO.
9BA8  AA                    TAX                     ; SAVE LSB OF BYTE COUNT.
9BA9  05DB                  ORA     MBC+1
9BAB  F01C ^9BC9            BEQ     :MI090          ; ZERO -- NOTHING TO DO.

9BAD  A000                  LDY     #0              ; INDEX TO DATA BLOCK.

9BAF  B1D6          :MI010  LDA     (MSP),Y         ; MOVE DATA.
9BB1  91D8                  STA     (MDP),Y
9BB3  C8                    INY                     ; BUMP INDEX.
9BB4  D004 ^9BBA            BNE     :MI020          ; NO PAGE WRAP.

9BB6  E6D7                  INC     MSP+1           ; PAGE WRAP -- BUMP POINTER VARIABLES.
9BB8  E6D9                  INC     MDP+1

9BBA  CA            :MI020  DEX                     ; DONE?
9BBB  D004 ^9BC1            BNE     :MI030          ; NO.

9BBD  A5DB                  LDA     MBC+1           ; NOT SURE -- CHECK FURTHER.
9BBF  F008 ^9BC9            BEQ     :MI090          ; YES -- DONE.

9BC1  E0FF          :MI030  CPX     #$FF            ; MAINTAIN D.P. BYTE COUNT.
9BC3  D0EA ^9BAF            BNE     :MI010

9BC5  C6DB                  DEC     MBC+1           ; BORROW FROM MSB.
9BC7  B0E6 ^9BAF            BCS     :MI010          ; (BRA).

9BC9  60            :MI090  RTS


9BCA                        PROC
                    ;
                    ; MOVDA -- MOVE DATA BLOCK WITH DECREASING ADDRESS
```

```
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       'MSP', 'MDP', & 'MBC' SETUP
                        ;       JSR     MOVDA
                        ;
98CA  A5DA              MOVDA   LDA     MBC                     ; SETUP BYTE COUNT ...
98CC  AA                        TAX
98CD  A8                        TAY                             ; ... AND DATA INDEX.
98CE  05DB                      ORA     MBC+1                   ; TEST FOR ZERO BYTE COUNT.
98D0  F024 ^98F6                BEQ     :MD090                  ; ZERO -- NOTHING TO DO.

98D2  18                        CLC                             ; ADJUST POINTERS FOR START.
98D3  A5D7                      LDA     MSP+1
98D5  65DB                      ADC     MBC+1
98D7  85D7                      STA     MSP+1

98D9  18                        CLC
98DA  A5D9                      LDA     MDP+1
98DC  65DB                      ADC     MBC+1
98DE  85D9                      STA     MDP+1

98E0  88                :MD010  DEY                             ; DECREMENT INDEX.
98E1  C0FF                      CPY     #$FF                    ; WRAP?
98E3  D006 ^98EB                BNE     :MD020                  ; NO.

98E5  C6DB                      DEC     MBC+1                   ; YES -- DECREMENT ALL POINTERS (MSB).
98E7  C6D7                      DEC     MSP+1
98E9  C6D9                      DEC     MDP+1

98EB  B1D6              :MD020  LDA     (MSP),Y                 ; MOVE A DATA BYTE.
98ED  91D8                      STA     (MDP),Y

98EF  CA                        DEX                             ; DONE?
98F0  D0EE ^98E0                BNE     :MD010                  ; NO -- CONTINUE.

98F2  A5DB                      LDA     MBC+1                   ; NOT SURE -- CHECK FURTHER.
98F4  D0EA ^98E0                BNE     :MD010                  ; NO -- CONTINUE.

98F6  60                :MD090  RTS                             ; YES -- RETURN.


98F7                            PROC
                        ;
                        ; MVINLN -- MOVE PART OF 'INLN' TO A FIXED ADDRESS BUFFER
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       Y = CURRENT INDEX IN 'INLN'
                        ;
                        ;       JSR :MINLN
                        ;
                        ;       'INLNBF' CONTAINS Y/Y+'INBFSZ'-1 CHARACTERS FROM 'INLN'
                        ;       LOWER CASE IS CONVERTED TO UPPER CASE.
                        ;       Y IS NOT PRESERVED.
                        ;
```

```
9BF7  A200        MVINLN  LDX     #0

9BF9  B180        :MVN10  LDA     (INLN),Y
9BFB  C9E1                CMP     #'A'+$20          ; LC?
9BFD  9006 ^9C05          BCC     :MVN20            ; NO.
9BFF  C97B                CMP     #'Z'+1+$20
9C01  B002 ^9C05          BCS     :MVN20            ; NO.
9C03  29DF                AND     #UC               ; YES -- LC -> UC.

9C05  9D3805      :MVN20  STA     INLNBF,X
9C08  C8                  INY
9C09  E8                  INX
9C0A  E00A                CPX     #INBFSZ
9C0C  90EB ^9BF9          BCC     :MVN10

9C0E  60                  RTS
```

```
                        ; DOUBLE PRECISION ROUTINES
                        ;
                        ; ALL VARIABLES ARE ACCESSED VIA THEIR OFFSET FROM SYMBOL 'DTAB'.
                        ; NORMALLY THE X AND/OR Y REGISTERS CONTAIN THE 'DTAB' OFFSET
                        ; VALUES TO THE VARIABLE(S) TO BE DEALT WITH.
                        ;


9C0F                            PROC
                        ;
                        ; DCMCI -- DOUBLE BYTE UNSIGNED COMPARE WITH CONSTANT.
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       X = DTAB OFFSET TO VARIABLE.
                        ;       Y = MSB OF CONSTANT.
                        ;       A = LSB OF CONSTANT.
                        ;
                        ;       JSR     DCMCI                   ; UNSIGNED COMPARE.
                        ;
                        ;       CC = DTAB(X) : Y,A
                        ;
9C0F  85A7              DCMCI   STA     TEMP2                   ; SAVE LSB.
9C11  84A8                      STY     TEMP2+1                 ; SAVE MSB.
9C13  A027                      LDY     #TEMP2-DTAB
                        ; *S*   JMP     DCMPI                   ; COMPARE & RETURN.


9C15                            PROC
                        ;
                        ; DCMPI -- DOUBLE BYTE UNSIGNED COMPARE INDEXED
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       X = DATA #1 OFFSET
                        ;       Y = DATA #2 OFFSET
                        ;
                        ;       JSR     DCMPI
                        ;       BEQ     DTAB(X) = DTAB(Y)
                        ;       BCS     DTAB(X) >= DTAB(Y)
                        ;       BCC     DTAB(X) < DTAB(Y)
                        ;
                        ;       CC = DTAB(X) : DTAB(Y)  (UNSIGNED)
                        ;
9C15  B5B1              DCMPI   LDA     DTAB+1,X                ; COMPARE MSBS.
9C17  D9B100                    CMP     DTAB+1,Y
9C1A  D005 ^9C21                BNE     :DC090                  ; NOT EQUAL -- ALL DONE.

                        ; *** EXTERNAL ENTRY POINT ***

9C1C  B5B0              DCM010  LDA     DTAB,X                  ; EQUAL -- COMPARE LSBS.
9C1E  D9B000                    CMP     DTAB,Y

9C21  60                :DC090  RTS
```

```
9C22                           PROC
                       ;
                       ; DSCMI -- DOUBLE BYTE SIGNED COMPARE INDEXED
                       ;
                       ; CALLING SEQUENCE:
                       ;
                       ;       X = DATA #1 OFFSET
                       ;       Y = DATA #2 OFFSET
                       ;
                       ;       JSR     DSCMI
                       ;       BEQ     DTAB(X) = DTAB(Y)
                       ;       BCS     DTAB(X) >= DTAB(Y)
                       ;       BCC     DTAB(X) < DTAB(Y)
                       ;
9C22  B96100           DSCMI   LDA     DTAB+1,Y        ; COMPARE MSBS FIRST.
9C25  4980                     EOR     #$80
9C27  85A1                     STA     TEMP
9C29  B581                     LDA     DTAB+1,X
9C2B  4980                     EOR     #$80
9C2D  C5A1                     CMP     TEMP
9C2F  F0EB ^9C1C               BEQ     DCM010          ; EQUAL -- COMPARE LSBS.

9C31  60                       RTS                     ; NOT EQUAL -- ALL DONE.



                       ;
                       ; DMOVI -- DOUBLE BYTE MOVE INDEXED
                       ;
                       ; CALLING SEQUENCE:
                       ;
                       ;       X = DESTINATION OFFSET
                       ;       Y = SOURCE OFFSET
                       ;
                       ;       JSR     DMOVI
                       ;
                       ;       DTAB(X) = DTAB(Y)
                       ;
                       ; *** SEE 'PMOVE' FOR THE 'DMOVI' CODE ***


9C32                           PROC
                       ;
                       ; DADDI -- DOUBLE PRECISION ADD
                       ;
                       ; CALLING SEQUENCE:
                       ;
                       ;       X = OFFSET TO
                       ;       Y = OFFSET TO
                       ;
                       ;       JSR     DADDI
                       ;       BVS     OVERFLOW
                       ;
```

```
                         ;       DTAB(X) = DTAB(X) + DTAB(Y)
                         ;
9C32  18                 DADDI   CLC

9C33  B580               DADDIX  LDA     DTAB,X
9C35  798000                     ADC     DTAB,Y
9C38  9580                       STA     DTAB,X

9C3A  B581                       LDA     DTAB+1,X
9C3C  798100                     ADC     DTAB+1,Y
9C3F  9581                       STA     DTAB+1,X

9C41  60                         RTS


9C42                             PROC
                         ;
                         ; DSUBI -- DOUBLE PRECISION SUBTRACT
                         ;
                         ; CALLING SEQUENCE:
                         ;
                         ;       X = OFFSET
                         ;       Y = OFFSET
                         ;
                         ;       JSR     DSUBI
                         ;       BVS     OVERFLOW
                         ;       BEQ     RESULT = 0
                         ;
                         ;       DTAB(X) = DTAB(X) - DTAB(Y)
                         ;
9C42  38                 DSUBI   SEC

9C43  B580               DSUBIX  LDA     DTAB,X
9C45  F98000                     SBC     DTAB,Y
9C48  9580                       STA     DTAB,X

9C4A  B581                       LDA     DTAB+1,X
9C4C  F98100                     SBC     DTAB+1,Y
9C4F  9581                       STA     DTAB+1,X

9C51  1580                       ORA     DTAB,X          ; SET CC FOR ZERO TEST.

9C53  60                         RTS


9C54                             PROC
                         ;
                         ; DMULI -- DOUBLE PRECISION MULTIPLY
                         ;
                         ; CALLING SEQUENCE:
                         ;
                         ;       Y = OFFSET
                         ;       X = OFFSET
                         ;
                         ;       JSR     DMULI
                         ;
                         ;      DTAB(X) = DTAB(X) * DTAB(Y)
                         ;
9C54  A910               DMULI   LDA    #16            ; SETUP LOOP COUNTER.
9C56  85A3                       STA    TEMP+2

9C58  A900                       LDA    #0             ; INITIALIZE TEMP ACCUMULATOR.
9C5A  85A1                       STA    TEMP
9C5C  85A2                       STA    TEMP+1

9C5E  1680               :DM010  ASL    DTAB,X         ; DOUBLE PRECISION SHIFT LEFT.
9C60  3681                       ROL    DTAB+1,X
9C62  900F ^9C73                 BCC    :DM020         ; NO BIT PRESENT.

9C64  18                         CLC                   ; BIT SET -- ADD TO PARTIAL.
9C65  A5A1                       LDA    TEMP
9C67  798000                     ADC    DTAB,Y
9C6A  85A1                       STA    TEMP
9C6C  A5A2                       LDA    TEMP+1
9C6E  798100                     ADC    DTAB+1,Y
9C71  85A2                       STA    TEMP+1

9C73  C6A3               :DM020  DEC    TEMP+2         ; DONE?
9C75  F007 ^9C7E                 BEQ    :DM090         ; YES -- RESULT IS IN 'TEMP'.

9C77  06A1                       ASL    TEMP           ; NO -- DOUBLE PRECISION SHIFT LEFT.
9C79  26A2                       ROL    TEMP+1
9C7B  4C5E9C                     JMP    :DM010

9C7E  A5A1               :DM090  LDA    TEMP           ; DONE -- MOVE RESULT.
9C80  9580                       STA    DTAB,X
9C82  A5A2                       LDA    TEMP+1
9C84  9581                       STA    DTAB+1,X
9C86  60                         RTS


9C87                             PROC
                         ;
                         ; DDIVI -- DOUBLE PRECISION DIVIDE
                         ;
                         ; CALLING SEQUENCE:
                         ;
                         ;       X = OFFSET TO DIVIDEND
                         ;       Y = OFFSET TO DIVISOR
                         ;
                         ;       JSR    DDIVI
                         ;
                         ;       DTAB(X) = DTAB(X) / DTAB(Y) (SIGNED)
                         ;       'TEMP' = REMAINDER (SIGN MAY BE WRONG!!!)
                         ;
9C87  B98000             DDIVI   LDA    DTAB,Y         ; CHECK FOR DIVIDE BY ZERO.
9C8A  198100                     ORA    DTAB+1,Y
9C8D  D005 ^9C94                 BNE    :DD003         ; NO -- O.K.

9C8F  A984                       LDA    #DIVERR        ; ERROR.
9C91  4C3A7A                     JMP    FSTOP
```

```
9C94  A911          :DD003  LDA    #15+1            ; SETUP LOOP COUNTER.
9C96  85A3                  STA    TEMP+2
9C98  86A4                  STX    TEMP+3           ; SAVE INDEX TO DIVIDEND.

9C9A  A900                  LDA    #0               ; INITIALIZE REMAINDER.
9C9C  85A1                  STA    TEMP
9C9E  85A2                  STA    TEMP+1

9CA0  B98100                LDA    DTAB+1,Y         ; SEE IF DIVISOR IS NEGATIVE.
9CA3  85A6                  STA    TEMP+5
9CA5  1006 ^9CAD            BPL    :DD006           ; NO.

9CA7  20F19C                JSR    DNEGI            ; YES -- NEGATE DIVIDEND ...
9CAA  20DD9C                JSR    :DD093           ; ... & DIVISOR (*** CRAZY CALL ***).

9CAD  B581          :DD006  LDA    DTAB+1,X         ; SEE IF DIVIDEND IS NEGATIVE.
9CAF  85A5                  STA    TEMP+4
9CB1  1003 ^9CB6            BPL    :DD008           ; NO.

9CB3  20F19C                JSR    DNEGI            ; YES -- NEGATE IT NOW (& THEN AGAIN LATER).

9CB6  18            :DD008  CLC

9CB7  A6A4          :DD010  LDX    TEMP+3           ; GET INDEX TO DIVIDEND.
9CB9  3680                  ROL    DTAB,X           ; DOUBLE PRECISION ROTATE.
9CBB  3681                  ROL    DTAB+1,X

9CBD  C6A3                  DEC    TEMP+2           ; DONE?
9CBF  F011 ^9CD2            BEQ    :DD090           ; YES.

9CC1  26A1                  ROL    TEMP             ; NO.
9CC3  26A2                  ROL    TEMP+1

9CC5  A221                  LDX    #TEMP-DTAB       ; IS REMAINDER < DIVISOR?
9CC7  20159C                JSR    DCMPI
9CCA  90EB ^9CB7            BCC    :DD010           ; YES.

9CCC  20429C                JSR    DSUBI            ; NO.
9CCF  38                    SEC
9CD0  B0E5 ^9CB7            BCS    :DD010           ; (BRA).

9CD2  A5A5          :DD090  LDA    TEMP+4           ; SEE IF RESULT IS TO BE NEGATED.
9CD4  1003 ^9CD9            BPL    :DD092           ; NO.

9CD6  20F19C                JSR    DNEGI            ; YES -- NEGATE POSITIVE RESULT.

9CD9  A5A6          :DD092  LDA    TEMP+5           ; WAS DIVISOR NEGATED EARLIER.
9CDB  1007 ^9CE4            BPL    :DD095           ; NO.

9CDD  98            :DD093  TYA                     ; YES -- NEGATE IT BACK TO ORIGINAL SIGN.
9CDE  AA                    TAX
9CDF  20F19C                JSR    DNEGI
9CE2  A6A4                  LDX    TEMP+3           ; RESTORE INDEX.

9CE4  60            :DD095  RTS
```

```
9CE5                      PROC
                    ;
                    ; DMODI -- MODULO OF SORTS
                    ;
                    ; CALLING SEQUENCE:
                    ;
                    ;       X = OFFSET TO DIVIDEND
                    ;       Y = OFFSET OT DIVISOR
                    ;
                    ;       JSR     DMODI
                    ;
                    ;       DTAB(X) = DTAB(X) MOD DTAB(Y)
                    ;
9CE5  20879C        DMODI   JSR     DDIVI            ; FIRST DO DIVISION.
9CE8  A5A1                  LDA     TEMP             ; TAKE ADVANTAGE OF SIDE EFFECT.
9CEA  9580                  STA     DTAB,X
9CEC  A5A2                  LDA     TEMP+1
9CEE  9581                  STA     DTAB+1,X
9CF0  60                    RTS

9CF1                        PROC
                    ;
                    ; DNEGI -- DOUBLE PRECISION NEGATE
                    ;
                    ; CALLING SEQUENCE:
                    ;
                    ;       X = OFFSET TO NUMBER
                    ;
                    ;       JSR     DNEGI
                    ;
                    ;       DTAB(X) = -DTAB(X)
                    ;
9CF1  38            DNEGI   SEC                      ; (CLEAR BORROW).
9CF2  A900                  LDA     #0
9CF4  F580                  SBC     DTAB,X
9CF6  9580                  STA     DTAB,X

9CF8  A900                  LDA     #0
9CFA  F581                  SBC     DTAB+1,X
9CFC  9581                  STA     DTAB+1,X
9CFE  60                    RTS


9CFF                        PROC
                    ;
                    ; DABSI -- DOUBLE PRECISION ABS FUNCTION
                    ;
                    ; CALLING SEQUENCE:
                    ;
                    ;       X = OFFSET TO NUMBER
                    ;
                    ;       JSR     DABSI
                    ;
                    ;       DTAB(X) = ABS (DTAB(X))
```

```
                          ;
9CFF  B581                DABSI   LDA     DTAB+1,X          ; CHECK SIGN OF MSB.
9D01  30EE ^9CF1                  BMI     DNEGI

9D03  60                          RTS


9D04                              PROC
                          ;
                          ; DADDS -- ADD A REGISTER TO DOUBLE BYTE
                          ;
                          ; CALLING SEQUENCE:
                          ;
                          ;         A = SIGNED BINARY NUMBER (-128 TO 127)
                          ;         X = DTAB OFFSET TO DP NUMBER
                          ;
                          ;         JSR     DADDS
                          ;
                          ;         DTAB(X) = DTAB(X) + A
                          ;
9D04  C900                DADDS   CMP     #0                ; SEE IF POSITIVE OR NEGATIVE.
9D06  300C ^9D14                  BMI     :DA030            ; NEGATIVE.

                          ; *** EXTERNAL ENTRY POINT ***

9D08  18                  DADDP   CLC                       ; POSITIVE -- ADD.
9D09  7580                        ADC     DTAB,X
9D0B  9580                        STA     DTAB,X
9D0D  9002 ^9D11                  BCC     :DA010            ; NO CARRY.

9D0F  F681                        INC     DTAB+1,X          ; CARRY -- ADD TO MSB.

9D11  60                  :DA010  RTS

                          ; *** EXTERNAL ENTRY POINT ***

9D12  A9FF                DDCRI   LDA     #-1

9D14  18                  :DA030  CLC
9D15  7580                        ADC     DTAB,X
9D17  9580                        STA     DTAB,X
9D19  B002 ^9D1D                  BCS     :DA040            ; NO BORROW.

9D1B  D681                        DEC     DTAB+1,X          ; BORROW -- SUB FROM MSB.

9D1D  60                  :DA040  RTS


9D1E                              PROC
                          ; RELATIONAL TESTS
                          ;
                          ; CALLING SEQUENCE:
                          ;
                          ;         X = DATA #1 OFFSET
```

```
                        ;       Y = DATA #2 OFFSET
                        ;
                        ;       JSR     DXXTI           ONE OF SIX ROUTINES
                        ;
                        ;       DTAB(X) = 1 IF RELATION TRUE, 0 IF FALSE

9D1E  20159C            DEQTI   JSR     DCMPI           ; UNSIGNED COMPARE (FASTER THAN SIGNED).
9D21  F027 ^9D4A                BEQ     DTRUE           ; EQUAL RESULTS IN TRUE.
9D23  D029 ^9D4E                BNE     DFALSE          ; UNEQUAL RESULTS IN FALSE.

9D25  20159C            DNETI   JSR     DCMPI           ; UNSIGNED COMPARE (FASTER THAN SIGNED).
9D28  D020 ^9D4A                BNE     DTRUE           ; UNEQUAL RESULTS IN TRUE.
9D2A  F022 ^9D4E                BEQ     DFALSE          ; EQUAL RESULTS IN FALSE.

9D2C  20229C            DGTTI   JSR     DSCMI           ; SIGNED COMPARE.
9D2F  F01D ^9D4E                BEQ     DFALSE          ; EQUAL RESULTS IN FALSE.
9D31  901B ^9D4E                BCC     DFALSE          ; LESS THAN RESULTS IN FALSE.
9D33  B015 ^9D4A                BCS     DTRUE           ; GREATER THAN RESULTS IN TRUE.

9D35  20229C            DLTTI   JSR     DSCMI           ; SIGNED COMPARE.
9D38  9010 ^9D4A                BCC     DTRUE           ; LESS THAN RESULTS IN TRUE.
9D3A  B012 ^9D4E                BCS     DFALSE          ; GREATER THAN OR EQUAL RESULTS IN FALSE.

9D3C  20229C            DGETI   JSR     DSCMI           ; SIGNED COMPARE.
9D3F  B009 ^9D4A                BCS     DTRUE           ; GREATER THAN OR EQUAL RESULTS IN TRUE.
9D41  9008 ^9D4E                BCC     DFALSE          ; LESS THAN RESULTS IN FALSE.

9D43  20229C            DLETI   JSR     DSCMI           ; SIGNED COMPARE.
9D46  F002 ^9D4A                BEQ     DTRUE           ; EQUAL RESULTS IN TRUE.
9D48  B004 ^9D4E                BCS     DFALSE          ; GREATER THAN RESULTS IN FALSE.
                        ;*S*    BCC     DTRUE           ; LESS THAN RESULTS IN TRUE.


9D4A  A901              DTRUE   LDA     #1              ;"TRUE" ...
9D4C  D002 ^9D50                BNE     DFA010          ; ... TO VARIABLE.

9D4E  A900              DFALSE  LDA     #0              ; "FALSE" ...

9D50  9580              DFA010  STA     DTAB,X          ; ... TO VARIABLE.
9D52  A900                      LDA     #0
9D54  9581                      STA     DTAB+1,X
9D56  60                        RTS


9D57                            PROC

      = 0000                    IF      LOGGRP
                        ;
                        ; DLANDI -- DOUBLE PRECISION LOGICAL AND
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       X = OFFSET
                        ;       Y = OFFSET
                        ;
                        ;       JSR     DLANDI
                        ;
                        ;       DTAB(X) = DTAB(X) LOGICAL AND DTAB(Y)
                        ;
     -                  DLANDI  JSR     DTXP                ; IS DTAB(X) FALSE?
     -                          BEQ     DFALSE              ; YES.

                        ; *** ENTRY FOR 'DLORI' ***

     -                  DAN010  JSR     DTYP                ; IS DTAB(Y) FALSE?
     -                          BEQ     DFALSE              ; YES -- SET DTAB(X) = FALSE AND EXIT.
     -                          BNE     DTRUE               ; NO -- SET DTAB(X) = TRUE AND EXIT.


     -                          PROC
                        ; DLORI -- DOUBLE PRECISION LOGICAL OR
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       X = OFFSET
                        ;       Y = OFFSET
                        ;
                        ;       JSR DLORI
                        ;
                        ;       DTAB(X) = DTAB(X) LOGICAL OR DTAB(Y)
                        ;
     -                  DLORI   JSR     DTXP                ; IS DTAB(X) TRUE?
     -                          BNE     DTRUE               ; YES.
     -                          BEQ     DAN010              ; NO (BRA).


                                ENDIF
 9D57                           PROC
                        ;
                        ; DLNOTI -- DOUBLE PRECISION LOGICAL NOT
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       X = OFFSET
                        ;
                        ;       JSR DLNOTI
                        ;
                        ;       DTAB(X) = LOGICAL NOT DTAB(X)
                        ;
 9D57   205E9D          DLNOTI  JSR     DTXP                ; TRUE OR FALSE?
 9D5A   F0EE ^9D4A              BEQ     DTRUE               ; FALSE -> TRUE AND EXIT.
 9D5C   D0F0 ^9D4E              BNE     DFALSE              ; TRUE -> FALSE AND EXIT.


 9D5E                           PROC
                        ;
                        ; DTXP -- DTAB(X) PREDICATE
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       X = OFFSET
                        ;
                        ;       JSR DTXP
                        ;
                        ;       BNE IF DTAB(X) POSITIVE (TRUE)
                        ;       BEQ IF DTAB(X) ZERO OR NEGATIVE (FALSE)
                        ;
9D5E  B581              DTXP    LDA     DTAB+1,X
9D60  3003 ^9D65                BMI     DTX010          ; NEGATIVE.
9D62  1580                      ORA     DTAB,X          ; POSITIVE OR ZERO.
9D64  60                        RTS                     ; CC IS SET.

9D65                    DTY010
9D65  A900              DTX010  LDA     #0
9D67  60                        RTS



                        ;
                        ; DANDI -- DOUBLE PRECISION AND
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       X = OFFSET
                        ;       Y = OFFSET
                        ;
                        ;       JSR     DANDI
                        ;
                        ;       DTAB(X) = DTAB(X) AND DTAB(Y)
                        ;
9D68  B580              DANDI   LDA     DTAB,X
9D6A  398000                    AND     DTAB,Y
9D6D  9580                      STA     DTAB,X
9D6F  B581                      LDA     DTAB+1,X
9D71  398100                    AND     DTAB+1,Y
9D74  9581                      STA     DTAB+1,X
9D76  60                        RTS



                        ;
                        ; DORI -- DOUBLE PRECISION OR
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       X = OFFSET
                        ;       Y = OFFSET
                        ;
                        ;       JSR     DORI
                        ;
                        ;       DTAB(X) = DTAB(X) OR DTAB(Y)
                        ;
9D77  B580              DORI    LDA     DTAB,X
9D79  198000                    ORA     DTAB,Y
9D7C  9580                      STA     DTAB,X
9D7E  B581                      LDA     DTAB+1,X
9D80  198100                    ORA     DTAB+1,Y
9D83  9581                      STA     DTAB+1,X
```

```
9085  60                  RTS


                  ;
                  ; DXORI -- DOUBLE PRECISION XOR
                  ;
                  ; CALLING SEQUENCE:
                  ;
                  ;       X = OFFSET
                  ;       Y = OFFSET
                  ;
                  ;       JSR     DXORI
                  ;
                  ;       DTAB(X) = DTAB(X) XOR DTAB(Y)
                  ;
9D86  B580        DXORI   LDA     DTAB,X
9D88  598000              EOR     DTAB,Y
9D8B  9580                STA     DTAB,X
9D8D  B581                LDA     DTAB+1,X
9D8F  598100              EOR     DTAB+1,Y
9D92  9581                STA     DTAB+1,X
9D94  60                  RTS


                  ;
                  ; DNOTI -- DOUBLE PRECISION NOT
                  ;
                  ; CALLING SEQUENCE:
                  ;
                  ;       X = OFFSET
                  ;
                  ;       JSR     DNOTI
                  ;
                  ;       DTAB(X) = NOT DTAB(X)
                  ;
9D95  B580        DNOTI   LDA     DTAB,X
9D97  49FF                EOR     #$FF
9D99  9580                STA     DTAB,X
9D9B  B581                LDA     DTAB+1,X
9D9D  49FF                EOR     #$FF
9D9F  9581                STA     DTAB+1,X
9DA1  60                  RTS


   = 0000                 IF      LOGGRF
 -                        PROC
                  ; DTYP -- DTAB(Y) PREDICATE.
                  ;
                  ; CALLING SEQUENCE:
                  ;
                  ;       Y = OFFSET
                  ;
                  ;       JSR DTYP
                  ;
```

```
                    ;       BNE IF DTAB(Y) POSITIVE (TRUE)
                    ;       BEQ IF DTAB(Y) ZERO OR NEGATIVE (FALSE)
                    ;
  -                 DTYP    LDA     DTAB+1,Y
  -                         BMI     DTY010          ; NEGATIVE.
  -                         ORA     DTAB,Y          ; POSITIVE OR ZERO.
  -                         RTS                     ; CC IS SET.
                            ENDIF
```

```
9DA2                    PROC
                ;
                ; ACCUMULATOR FUNCTIONS -- ASSUME THE EXISTENCE OF A DOUBLE PRECISION
                ; VARIABLE WITHIN 'DTAB' NAMED 'ACC'.
                ;

                ;
                ; DLOADA -- LOAD 'ACC' WITH DATA
                ;
                ; CALLING SEQUENCE:
                ;
                ;       Y = OFFSET TO SOURCE DATA
                ;
                ;       JSR     DLOADA
                ;
                ;       X = ACC OFFSET
                ;       'ACC' = DTAB(Y)
                ;
9DA2  A262      DLOADA  LDX     #ACC-DTAB
9DA4  4C459A            JMP     DMOVI

9DA7                    PROC
                ;
                ; DSTORA -- STORE 'ACC' TO LOCATION
                ;
                ; CALLING SEQUENCE:
                ;
                ;       X = OFFSET TO DESTINATION
                ;
                ;       JSR     DSTORA
                ;
                ;       Y = 'ACC' OFFSET
                ;       DTAB(X) = 'ACC'
                ;
9DA7  A062      DSTORA  LDY     #ACC-DTAB
9DA9  4C459A            JMP     DMOVI

9DAC                    PROC
                ;
                ; DADDA -- ADD DATA TO 'ACC'
                ;
                ; CALLING SEQUENCE:
                ;
                ;       Y = OFFSET TO DATA
                ;
                ;       JSR     DADDA
                ;
                ;       X = 'ACC' OFFSET
                ;       'ACC' = 'ACC' + DTAB(Y)
                ;
9DAC  A262      DADDA   LDX     #ACC-DTAB
9DAE  4C329C            JMP     DADDI

9DB1                    PROC
```

```
9D81                    PROC
```

```
                  ;
                  ; DSUBA -- SUBTRACT DATA FROM 'ACC'
                  ;
                  ; CALLING SEQUENCE:
                  ;
                  ;       Y = OFFSET TO DATA
                  ;
                  ;       JSR     DSUBA
                  ;       BEQ     RESULT = 0
                  ;
                  ;       X = 'ACC' OFFSET
                  ;       'ACC' = 'ACC' - DTAB(Y)
                  ;
9D81  A262        DSUBA   LDX     #ACC-DTAB
9D83  4C429C              JMP     DSUBI


9D86                      PROC
                  ;
                  ; DCMPA -- COMPARE 'ACC' WITH DATA (UNSIGNED)
                  ;
                  ; CALLING SEQUENCE:
                  ;
                  ;       Y = DATA OFFSET
                  ;
                  ;       JSR     DCMPA
                  ;
                  ;       CC = 'ACC' : DTAB(Y)  (UNSIGNED)
                  ;       X = 'ACC' OFFSET
                  ;
9D86  A262        DCMPA   LDX     #ACC-DTAB
9D88  4C159C              JMP     DCMPI
```

```
9DBB                          PROC
                       ;
                       ; ASCDEC -- DECIMAL IN ASCII TO BINARY CONVERSION
                       ;
                       ; CALLING SEQUENCE:
                       ;
                       ;        X = DTAB OFFSET TO POINTER VARIABLE
                       ;        Y = OFFSET WITHIN STRING TO START OF NUMBER
                       ;
                       ;        JSR      ASCDEC
                       ;
                       ;        'NUMBER' = RESULT OF CONVERSION (MODULO 2**16)
                       ;        Y = INDEX TO END OF NUMBER DELIMITER
                       ;        USES 'TEMP' THRU 'TEMP'+4
                       ;
 9DBB  A900            ASCDEC  LDA     #0                 ; INITIALIZE RESULT.
 9DBD  85B8                    STA     NUMBER
 9DBF  85B9                    STA     NUMBER+1

 9DC1  B580                    LDA     DTAB,X             ; MOVE POINTER.
 9DC3  85A3                    STA     TEMP+2
 9DC5  B581                    LDA     DTAB+1,X
 9DC7  85A4                    STA     TEMP+3
 9DC9  B583                    LDA     DTAB+3,X           ; SAVE END INDEX.
 9DCB  85A5                    STA     TEMP+4

 9DCD  B1A3                    LDA     (TEMP+2),Y
 9DCF  C92D                    CMP     #'-'                ; UNARY MINUS?
 9DD1  D009 ^9DDC              BNE     :AC010             ; NO.

 9DD3  C8                      INY                        ; YES -- SKIP OVER IT.
 9DD4  20DC9D                  JSR     :AC010             ; *** RECURSIVE CALL ***.
 9DD7  A238                    LDX     #NUMBER-DTAB
 9DD9  4CF19C                  JMP     DNEGI              ; NEGATE RESULT & RETURN.

 9DDC  C4A5            :AC010  CPY     TEMP+4             ; END OF STRING?
 9DDE  F033 ^9E13              BEQ     :AC090             ; YES.

 9DE0  B1A3                    LDA     (TEMP+2),Y         ; GET A CHARACTER.
 9DE2  20839E                  JSR     CNUMBR             ; VALID DECIMAL DIGIT?
 9DE5  B02C ^9E13              BCS     :AC090             ; NO -- DONE.

 9DE7  C8                      INY
 9DE8  48                      PHA                        ; YES -- SAVE IT.
 9DE9  06B8                    ASL     NUMBER             ; X2.
 9DEB  26B9                    ROL     NUMBER+1

 9DED  A5B9                    LDA     NUMBER+1           ; SAVE X2.
 9DEF  85A2                    STA     TEMP+1
 9DF1  A5B8                    LDA     NUMBER
 9DF3  85A1                    STA     TEMP

 9DF5  0A                      ASL     A                  ; X4.
 9DF6  26B9                    ROL     NUMBER+1

 9DF8  0A                      ASL     A                  ; X8.
 9DF9  26B9                    ROL     NUMBER+1
```

```
9DFB  18                     CLC                          ; X10 = X8 + X2.
9DFC  65A1                   ADC      TEMP
9DFE  85B8                   STA      NUMBER
9E00  9003 ^9E05             BCC      :AC020              ; NO CARRY.

9E02  E6B9                   INC      NUMBER+1            ; CARRY -- ADD TO MSB.
9E04  18                     CLC


9E05  68            :AC020   PLA                          ; GET NEW DIGIT.
9E06  65B8                   ADC      NUMBER              ; ADD TO PARTIAL RESULT.
9E08  85B8                   STA      NUMBER
9E0A  A5B9                   LDA      NUMBER+1
9E0C  65A2                   ADC      TEMP+1
9E0E  85B9                   STA      NUMBER+1
9E10  4CDC9D                 JMP      :AC010

9E13  60            :AC090   RTS


9E14                         PROC
                    ;
                    ; DECASC -- BINARY TO DECIMAL IN ASCII CONVERSION
                    ;
                    ; CALLING SEQUENCE:
                    ;
                    ;        X = DTAB INDEX TO SIGNED VALUE
                    ;
                    ;        JSR     DECASC
                    ;
                    ;        PRINTS RESULT TO 'CHOT' ROUTINE
                    ;        USES 'TEMP'+2 THRU 'TEMP'+5 & 'TEMP2' THRU 'TEMP2'+2
                    ;
9E14  84A6          DECASC   STY      TEMP+5              ; SAVE Y REGISTER.
9E16  B580                   LDA      DTAB,X              ; MOVE DATA TO TEMPORARY STORAGE.
9E18  85A7                   STA      TEMP2
9E1A  B581                   LDA      DTAB+1,X
9E1C  85A8                   STA      TEMP2+1
9E1E  100A ^9E2A             BPL      :DC020              ; NUMBER IS POSITIVE.

9E20  A227                   LDX      #TEMP2-DTAB         ; NEGATE NUMBER.
9E22  20F19C                 JSR      DNEGI
9E25  A92D                   LDA      #'-'                ; PRINT LEADING MINUS SIGN.
9E27  208294                 JSR      CHOT                ; PRINT A CHARACTER.

9E2A  A000          :DC020   LDY      #0                  ; INITIALIZE CONVERSION INDEX ...
9E2C  84A9                   STY      TEMP2+2             ; ... & LEADING ZERO SUPPRESS FLAG.

9E2E  B9799E        :DC030   LDA      PTEN,Y              ; GET POWER OF TEN.
9E31  85A3                   STA      TEMP+2
9E33  B97A9E                 LDA      PTEN+1,Y
9E36  85A4                   STA      TEMP+3
9E38  84A5                   STY      TEMP+4              ; SAVE INDEX TO TABLE.

9E3A  A930                   LDA      #'0'                ; INITIALIZE DIGIT.
9E3C  8D1105                 STA      DIGIT
```

```
                   ATARI CAMAC Assembler Ver  1.0A  Page 187
PILOT -- H.G. STEWART                                    D1:PILOT.


9E3F  A237                    LDX     #TEMP2-DTAB       ; PREPARE FOR SUCCESSIVE SUBTRACTION.
9E41  A025                    LDY     #TEMP+2-DTAB

9E43  20429C        :DC040    JSR     DSUBI
9E46  A548                    LDA     TEMP2+1           ; SEE IF RESULT IS NEGATIVE.
9E48  3005 ^9E4F              BMI     :DC045            ; YES -- ENOUGH ALREADY.

9E4A  EE1105                  INC     DIGIT             ; NO -- KEEP SUBTRACTING.
9E4D  D0F4 ^9E43              BNE     :DC040            ; (BRA).

9E4F  20329C        :DC045    JSR     DADCI             ; NOW CORRECT FROM ONE TOO MANY SUBTRACTS.

9E52  A549                    LDA     TEMP2+2           ; SEE IF NON-ZERO DIGIT HAS BEEN PRINTED YET.
9E54  D009 ^9E5F              BNE     :DC050            ; YES -- PRINT ALL SUBSEQUENT DIGITS.

9E56  AD1105                  LDA     DIGIT             ; NO -- SEE IF THIS DIGIT IS ANOTHER ZERO.
9E59  C930                    CMP     #'0'
9E5B  F008 ^9E65              BEQ     :DC060            ; YES IT IS -- SUPPRESS IT.

9E5D  8549                    STA     TEMP2+2           ; NO -- SET FLAG AND PRINT DIGIT.

9E5F  AD1105        :DC050    LDA     DIGIT             ; PRINT DIGIT.
9E62  208294                  JSR     CHOT

9E65  A4A5          :DC060    LDY     TEMP+4            ; RESTORE TABLE INDEX.
9E67  C8                      INY
9E68  C8                      INY
9E69  C00A                    CPY     #PTENL            ; DONE?
9E6B  D0C1 ^9E2E              BNE     :DC030            ; NO.

9E6D  A549                    LDA     TEMP2+2           ; WAS THE NUMBER = 0?
9E6F  D005 ^9E76              BNE     :DC070            ; NO.

9E71  A930                    LDA     #'0'               ; YES -- PRINT SINGLE ZERO DIGIT.
9E73  208294                  JSR     CHOT

9E76  A4A6          :DC070    LDY     TEMP+5            ; YES -- RESTORE Y REGISTER ...
9E78  60                      RTS                       ; ... & RETURN.


9E79  1027E80364  PTEN     DW    10000,1000,100,10,1      ; DECREASING POWERS OF TEN.
       = 000A     PTENL = *-PTEN                                ; TABLE LENGTH IN WORDS.
```

```
9E83                       PROC
                   ;
                   ; CNUMBR -- CHECK ASCII CHARACTER FOR VALID NUMBER ('0 - '9)
                   ;
                   ; CALLING SEQUENCE:
                   ;
                   ;       A = ASCII CHARACTER
                   ;
                   ;       JSR     CNUMBR
                   ;       BCS     NOT DECIMAL DIGIT
                   ;
                   ;       A = BINARY DIGIT
                   ;
9E83  C930         CNUMBR  CMP     #'0'            ; < '0?
9E85  9004 ^9E8B           BCC     :CN010          ; YES -- INVALID.

9E87  C93A                 CMP     #'9'+1          ; > '9?
9E89  9002 ^9E8D           BCC     :CN020          ; NO -- VALID DECIMAL DIGIT.

9E8B  38           :CN010  SEC                     ; SET CARRY FOR EXIT.
9E8C  60                   RTS

9E8D  E92F         :CN020  SBC     #'0'-1          ; (ADJUST FOR CARRY CLEAR).
9E8F  18                   CLC                     ; SET CC FOR EXIT.
9E90  60                   RTS


9E91                       PROC
                   ;
                   ; CLETTR -- CHECK ASCII CHARACTER FOR ALPHA LETTER ('A - 'Z)
                   ;
                   ; CALLING SEQUENCE:
                   ;
                   ;       A = ASCII CHARACTER
                   ;
                   ;       JSR     CLETTR
                   ;       BCS     NOT ALPHA LETTER
                   ;
                   ;       A = ASCII CHARACTER
                   ;
9E91  48           CLETTR  PHA                     ; SAVE CHARACTER.
9E92  29DF                 AND     #UC             ; FORCE UPPER CASE.
9E94  C941                 CMP     #'A'            ; < 'A?
9E96  9004 ^9E9C           BCC     :CL010          ; YES -- NOT ALPHA.

9E98  C95B                 CMP     #'Z'+1          ; > 'Z?
9E9A  9001 ^9E9D           BCC     :CL020          ; NO -- VALID LETTER.

9E9C  38           :CL010  SEC                     ; SET CARRY FOR EXIT.

9E9D  68           :CL020  PLA                     ; RESTORE CHARACTER.
9E9E  60                   RTS
```

```
 9E9F                        PROC
                     ;
                     ; STMLST -- SETUP LIST POINTER TO STATEMENT LIST
                     ;
 9E9F                STMLST
 9E9F  A5AE                  LDA     S1L             ; 'LP' = 'S1L'.
 9EA1  85BA                  STA     LP
 9EA3  A5AF                  LDA     S1L+1
 9EA5  85BB                  STA     LP+1
 9EA7  A900                  LDA     #ATRLIN         ; 'LIN' FOR LINE # FIND.
 9EA9  8D6605                STA     ATRTYP
 9EAC  60                    RTS


 9EAD                        PROC
                     ;
                     ; SETSVL -- SETUP LIST POINTER TO NAMED STRING LIST
                     ;
 9EAD  A5B2          SETSVL  LDA     S2L             ; 'LP' = 'S2L'.
 9EAF  85BA                  STA     LP
 9EB1  A5B3                  LDA     S2L+1
 9EB3  85BB                  STA     LP+1
 9EB5  60                    RTS


 9EB6                        PROC
                     ;
                     ; CKEOA -- CHECK FOR END OF ATOM (NON-ALPHANUMERIC CHARACTER)
                     ;
                     ; CALLING SEQUENCE:
                     ;
                     ;       A = ASCII CHARACTER
                     ;
                     ;       JSR     CKEOA
                     ;       BEQ     END OF ATOM (NOT AN ALPHANUMERIC CHARACTER)
                     ;
 9EB6  20919E        CKEOA   JSR     CLETTR          ; ALPHA LETTER?
 9EB9  900C ^9EC7            BCC     :CK090          ; YES.

 9EBB  48                    PHA
 9EBC  20B39E                JSR     CNUMBR          ; NO -- NUMERIC CHARACTER?
 9EBF  68                    PLA
 9EC0  9005 ^9EC7            BCC     :CK090          ; YES.

 9EC2  85A1                  STA     TEMP            ; NEITHER -- SET CC FOR EXIT.
 9EC4  C5A1                  CMP     TEMP
 9EC6  60                    RTS

 9EC7  C9FF          :CK090  CMP     #$FF            ; SET CC FOR EXIT.
 9EC9  60                    RTS
```

```
                        ;
                        ; SCEOA -- SCAN TO END OF ATOM
                        ;
9ECA  C8                        INY

9ECB  B180              SCEOA   LDA     (INLN),Y
9ECD  20B69E                    JSR     CKECA           ; END OF ATOM?
9ED0  D0F8 ^9ECA                BNE     SCECA-1         ; NO.

9ED2  60                        RTS                     ; YES -- RETURN WITH CC SET.


9ED3                            PROC
                        ;
                        ; SCNLBL -- IDENTIFY (& SCAN TO END OF) LABEL
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       Y = INDEX TO INPUT LINE.
                        ;
                        ;       JSR     SCNLBL
                        ;       BNE     NO LABEL PRESENT (A = CODE).
                        ;
                        ;       Y = INDEX TO END OF LABEL + 1
                        ;  INDENT = INDEX TO FIRST NON-SEPARATOR.
                        ;
                        ; NOTE: JUMPS TO 'PSTOP' IF INVALID LABEL NAME FOUND.
                        ;
9ED3  20079F            SCNLBL  JSR     SKPSEP          ; SKIP LEADING BLANKS AND/OR COMMAS.
9ED6  8CCA05                    STY     INDENT          ; UPDATE 'AUTO INDENT'.
9ED9  C92A                      CMP     #'*'            ; LABEL PREFIX DELIMITER?
9EDB  F003 ^9EE0                BEQ     :SL005          ; YES.

9EDD  A902                      LDA     #IMPERR         ; NO LABEL.
9EDF  60                        RTS

9EE0  C8                :SL005  INY
9EE1  B180                      LDA     (INLN),Y
9EE3  20B69E                    JSR     CKECA           ; SEE IF AT LEAST ONE ALPHANUMERIC.
9EE6  D0E3 ^9ECB                BNE     SCECA           ; YES -- SCAN TO END OF ATOM & RETURN.

9EE8  A902                      LDA     #ATMERR         ; NO -- INVALID LABEL NAME.
9EEA  4C3A7A                    JMP     PSTOP


9EED                            PROC
                        ;
                        ; CHKSEP -- CHECK FOR OPERAND SEPARATOR CHARACTER
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       A = CHARACTER.
                        ;
                        ;       JSR     CHKSEP
                        ;       BNE     NOT A SEPARATOR
```

```
                          ;
9EED  C920              CHKSEP  CMP     #' '              ; BLANK?
9EEF  F007 ^9EF8                BEQ     :CS090            ; YES.

9EF1  C92C                      CMP     #','              ; COMMA?
9EF3  F003 ^9EF8                BEQ     :CS090            ; YES.

9EF5  4CF99E                    JMP     CHKTRM            ; END OF STATEMENT CHECK & RETURN.

9EF8  60                :CS090  RTS



9EF9                            PROC
                          ;
                          ; CHKTRM -- CHECK FOR STATEMENT TERMINATOR (EOL OR '[').
                          ;
                          ; CALLING SEQUENCE:
                          ;
                          ;       A = CHARACTER.
                          ;
                          ;       JSR     CHKTRM
                          ;       BNE     NOT STATEMENT TERMINATOR.
                          ;
9EF9  C99B              CHKTRM  CMP     #EOL
9EFB  F002 ^9EFF                BEQ     :CK090

9EFD  C95B                      CMP     #SBRACK

9EFF  60                :CK090  RTS



9F00                            PROC
                          ;
                          ; CHKEQS -- CHECK FOR EQUAL SIGN
                          ;
                          ; CALLING SEQUENCE:
                          ;
                          ;       Y = 'INLN' INDEX.
                          ;
                          ;       JSR     CHKEQS
                          ;       BEQ     1ST NON-BLANK CHARACTER WAS '='.
                          ;
                          ;       Y = 'INLN' INDEX TO 1ST NON-BLANK CHAR.
                          ;
9F00  20139F            CHKEQS  JSR     SLB               ; SKIP LEADING BLANKS.
9F03  C93D                      CMP     #'='
9F05  60                        RTS                       ; RETURN WITH CC SET.



9F06                            PROC
                          ;
```

```
                        ; SKPSEP -- SKIP OPERAND SEPARATOR(S)
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       Y = INDEX TO INPUT LINE
                        ;
                        ;       JSR     SKPSEP
                        ;
                        ;       Y = INDEX TO FIRST NON-SEPARATOR FOUND
                        ;
                        ; NOTE: ANY STRING OF CONSECUTIVE BLANKS AND/OR COMMAS IS TREATED AS A SINGLE
                        ;       SEPARATOR.
                        ;
9F06  C8                        INY

9F07  B180              SKPSEP  LDA     (INLN),Y
9F09  C920                      CMP     #' '            ; BLANK?
9F0B  F0F9 ^9F06                BEQ     SKPSEP-1        ; YES.

9F0D  C92C                      CMP     #','            ; COMMA?
9F0F  F0F5 ^9F06                BEQ     SKPSEP-1        ; YES.

9F11  60                        RTS



9F12                            PROC
                        ;
                        ; SLB -- SKIP LEADING BLANKS
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       JSR     SLB
                        ;
                        ;       A = FIRST NON-BLANK CHARACTER FOUND.
                        ;
9F12  C8                        INY

9F13  B180              SLB     LDA     (INLN),Y
9F15  C920                      CMP     #' '            ; BLANK?
9F17  F0F9 ^9F12                BEQ     SLB-1           ; YES -- KEEP SCANNING.

9F19  60                        RTS



9F1A                            PROC
                        ;
                        ; SCNEOL -- SCAN TO END OF LINE
                        ;
9F1A  C8                        INY

9F1B  B180              SCNEOL  LDA     (INLN),Y
9F1D  C99B                      CMP     #EOL
9F1F  D0F9 ^9F1A                BNE     SCNEOL-1
```

```
 9F21  60                 RTS                   ; RETURN WITH CC SET.


 9F22                     PROC
                 ;
                 ; PSF -- PRINT A STORAGE FORMAT LINE
                 ;
                 ; CALLING SEQUENCE:
                 ;
                 ;        Y = INDEX TO LINE POINTER.
                 ;
                 ;        JSR     PSF
                 ;
 9F22  A236      PSF      LDX     #POINT-DTAB    ; MOVE POINTER TO 'POINT'.
 9F24  20459A             JSR     DMOVI
 9F27  208C9F             JSR     GTLNNO         ; GET LINE # TO 'LINENO'.

 9F2A  A6DD               LDX     LINENO+1       ; LEADING SPACES TO RIGHT-JUSTIFY LINE #.
 9F2C  E003               CPX     # HIGH 1000    ; >= 1000?
 9F2E  9008 ^9F38         BCC     :PS002         ; NO.
 9F30  D01C ^9F4E         BNE     :PS003         ; YES.
 9F32  A5DC               LDA     LINENO
 9F34  C9E8               CMP     # LOW 1000
 9F36  B016 ^9F4E         BCS     :PS003         ; YES.
 9F38  20A29F    :PS002   JSR     SPACE

 9F3B  8A                 TXA                    ; >= 100?
 9F3C  D010 ^9F4E         BNE     :PS003         ; YES.
 9F3E  A6DC               LDX     LINENO
 9F40  E064               CPX     # 100          ; >= 100?
 9F42  B00A ^9F4E         BCS     :PS003         ; YES.
 9F44  20A29F             JSR     SPACE

 9F47  E00A               CPX     # 10           ; >= 10?
 9F49  B003 ^9F4E         BCS     :PS003         ; YES.
 9F4B  20A29F             JSR     SPACE

 9F4E  A25C      :PS003   LDX     #LINENO-DTAB
 9F50  20149E             JSR     DECASC         ; PRINT BINARY LINE #.

 9F53  C8                 INY                    ; LOOK AHEAD TO 1ST CHAR OF STATEMENT.
 9F54  C8                 INY
 9F55  A920               LDA     #' '           ; IS IT A SPACE?
 9F57  D1B6               CMP     (POINT),Y
 9F59  F003 ^9F5E         BEQ     :PS005         ; YES.

 9F5B  20A29F             JSR     SPACE          ; NO -- PUT SPACE BETWEEN LINE # AND STATEMENT.

 9F5E  88        :PS005   DEY                    ; GET STATEMENT LENGTH.
 9F5F  B1B6               LDA     (POINT),Y
 9F61  AA                 TAX

 9F62  CEFE02             DEC     DSPFLG         ; DISPLAY CONTROL CHARACTERS.

 9F65  C8        :PS010   INY                    ; PRINT STATEMENT BODY.
 9F66  B1B6               LDA     (POINT),Y
```

```
9F68  208294              JSR     CHOT
9F6B  CA                  DEX
9F6C  D0F7 ^9F65          BNE     :PS010

9F6E  EEFE02              INC     DSPFLG          ; BACK TO ZERO.

9F71  60                  RTS



9F72                      PROC
                    ;
                    ; NULACC -- SET THE ACCEPT BUFFER TO NULL (SINGLE SPACE)
                    ;
9F72  A000          NULACC LDY    #0
9F74  A920                LDA     #' '             ; SINGLE SPACE.
9F76  9188                STA     (ACLN),Y
9F78  848A                STY     ACLN+2          ; START INDEX.
9F7A  C8                  INY
9F7B  848B                STY     ACLN+3          ; END INDEX.
9F7D  60                  RTS




9F7E                      PROC
                    ;
                    ; ABRTCK -- BREAK KEY ABORT CHECK
                    ;
9F7E  48            ABRTCK PHA                    ; (SEE 'XSYNC' ).
9F7F  A511                LDA     BREAK           ; OPERATOR ABORT?
9F81  D007 ^9F8A          BNE     :AC090          ; NO.

9F83  C611                DEC     BREAK           ; YES -- RESET FLAG.

9F85  A987          :AC005 LDA    #ABTERR         ; STOP WITH STATUS CODE.
9F87  4C3A7A              JMP     PSTOP

      = 0000              IF      FALSE
   -                :AC010 LDA    CONKEY          ; ALTERNATE ABORT?
   -                      AND     #STRTKY         ; START KEY?
   -                      BEQ     :AC090          ; NO.

   -                      LDA     CONKEY          ; YES -- RESET STATUS.
   -                      AND     #$FF-STRTKY
   -                      STA     CONKEY
   -                      JMP     :AC005
                          ENDIF

9F8A  68            :AC090 PLA
9F8B  60                  RTS

9F8C                      PROC
                    ;
                    ; GTLNNO -- GET LINE # FROM STORAGE LINE
```

```
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       'POINT' POINTS TO STORAGE LINE
                        ;
                        ;       JSR     GTLNNO
                        ;
                        ;       'LINENO' = BINARY LINE #
                        ;       Y = 4
                        ;
9F8C  A003              GTLNNO  LDY     #3
9F8E  B186                      LDA     (POINT),Y
9F90  850D                      STA     LINENO+1        ; RE-INVERT ORDER.
9F92  C8                        INY
9F93  B186                      LDA     (POINT),Y
9F95  850C                      STA     LINENO
9F97  60                        RTS


9F98                            PROC
                        ; NEWLIN -- ISSUE NEW LINE SEQUENCE TO 'CHOT'

9F98  A99B              NEWLIN  LDA     #EOL
9F9A  4C8294                    JMP     CHOT            ; NEWLINE & RETURN.



9F9D                            PROC
                        ; SPACE(S) -- ISSUE SPACE(S) TO 'CHOT'

9F9D  A920              SPACES  LDA     #' '            ; TWO SPACES.
9F9F  208294                    JSR     CHOT

9FA2  A920              SPACE   LDA     #' '            ; ONE SPACE.
9FA4  4C8294                    JMP     CHOT            ; & RETURN.


9FA7                            PROC
                        ; CRSNOP -- COMPLICATED NOP TO UPDATE CURSOR INHIBIT/ENABLE STATE
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       A = 0 TO ENABLE CURSOR, ELSE DISABLE CURSOR.
                        ;

9FA7  8DF002            CRSNOP  STA     CRSINH          ; SET CURSOR INHIBIT FLAG.
9FAA  A91C                      LDA     #CUP            ; CURSOR UP ...
9FAC  208294                    JSR     CHOT
9FAF  A91D                      LDA     #CDOWN          ; ... THEN DOWN ...
9FB1  4C8294                    JMP     CHOT            ; ... & RETURN.


9FB4                            PROC
```

```
                     ;
                     ; AUDCLR -- CLEAR AUDIO REGISTERS AND SELECTS
                     ;
9FB4  A903           AUDCLR  LDA    #$03          ; MAGIC CONSTANT FROM D. CRANE, 27-AUG-79.
9FB6  8D3202                 STA    SSKCTL
9FB9  8D0FD2                 STA    SKCTL

9FBC  A900                   LDA    #0
9FBE  8D08D2                 STA    AUDCTL        ; SET AUDIO TO 4 INDEPENDENT REGISTERS.

9FC1  A208                   LDX    #AUREGS*2

9FC3  9DFED1         :AC010  STA    AUDF1-2,X     ; CLEAR ALL ACTIVE TONES.
9FC6  9DFFD1                 STA    AUDC1-2,X
9FC9  9D1305                 STA    AUDIOR-2,X    ; CLEAR 'SO' SELECTS.
9FCC  9D1405                 STA    AUDIOR-1,X
9FCF  CA                     DEX
9FD0  CA                     DEX
9FD1  D0F0 ^9FC3             BNE    :AC010

9FD3  60                     RTS
```

```
9FD4                    PROC
                ;
                ; EXP -- ARITHMETIC EXPRESSION EVALUATOR
                ;
                ; CALLING SEQUENCE:
                ;
                ;        'INLN' POINTS TO LINE TO BE EVALUATED
                ;        Y = INDEX TO START OF EXPRESSION
                ;
                ;        JSR     EXP
                ;
                ;        Y = INDEX TO END OF EXPRESSION + 1
                ;        'EXPSTK'+0 & +1 = RESULT OF EVALUATION.
                ;
9FD4  A900      EXP     LDA     #0              ; INITIALIZE CRITICAL VARIABLES.
9FD6  8D4E05            STA     ESTKP

                ; *** EXTERNAL ENTRY POINT ***

9FD9  2014A0    EXPRC   JSR     EXPVAL          ; CHECK FOR OPERAND & GET VALUE TO STACK.

9FDC  20139F    :EX030  JSR     SLB             ; SKIP LEADING BLANKS.
9FDF  84A7              STY     TEMP2           ; SAVE INDEX.

9FE1  206E81            JSR     ATOM            ; CHECK FOR OPERATOR.
9FE4  D021 ^A007        BNE     :EX080          ; INVALID ATOM.

9FE6  C940              CMP     #OPR
9FE8  D01D ^A007        BNE     :EX080          ; NOT AN OPERATOR.

9FEA  AE4E05            LDX     ESTKP           ; PUSH OPERATOR ROUTINE ADDR TO EXP STACK.
9FED  E00E              CPX     #ESTKSZ
9FEF  F019 ^A00A        BEQ     EXP192          ; STACK FULL.

9FF1  A586              LDA     POINT
9FF3  9593              STA     EXPSTK,X
9FF5  A587              LDA     POINT+1
9FF7  9594              STA     EXPSTK+1,X
9FF9  E8                INX
9FFA  E8                INX
9FFB  8E4E05            STX     ESTKP

9FFE  2014A0            JSR     EXPVAL          ; CHECK FOR OPERAND & GET VALUE TO STACK.

A001  207CA0            JSR     SOP             ; OPERATE ON STACK DATA.
A004  4CDC9F            JMP     :EX030

A007  A4A7      :EX080  LDY     TEMP2
A009  60                RTS

A00A  A902      EXP192  LDA     #EXPERR

A00C  4C3A7A    EXP194  JMP     PSTOP


A00F                    PROC
```

```
                            ;
                            ; EXPVAL -- VALIDATE OPERAND & PUSH VALUE TO STACK
                            ;
                            ; CALLING SEQUENCE:
                            ;
                            ;       Y = INDEX TO 'INLN'
                            ;
                            ;       JSR     EXPVAL
                            ;
A00F  A900                  EXPP    LDA     #0              ; EVALUATE EXPR IN PARENS.
A011  8D4E05                        STA     ESTKP

A014  20139F                EXPVAL  JSR     SLB
A017  A202                          LDX     #UNTABX         ; UNARY OPERATOR?
A019  20A87C                        JSR     SBCMAT
A01C  D01D ^A03B                    BNE     :EX010          ; NO.

A01E  8A                            TXA                     ; YES.
A01F  48                            PHA                     ; SAVE OFFSET IN 'SBDTAB'.
A020  2014A0                        JSR     EXPVAL          ; *** RECURSIVE CALL ***

A023  68                            PLA                     ; RESTORE OFFSET IN 'SBDTAB'.
A024  AA                            TAX
A025  BD4A80                        LDA     SBDTAB,X        ; GET OPERATOR ROUTINE ADDRESS.
A028  8D0E05                        STA     SJUMP+1
A02B  BD4B80                        LDA     SBDTAB+1,X
A02E  8D0F05                        STA     SJUMP+2

A031  AD4E05                        LDA     ESTKP           ; GET OFFSET TO RESULT.
A034  18                            CLC
A035  6911                          ADC     #EXPSTK-DTAB-2
A037  AA                            TAX
A038  4C0D05                        JMP     SJUMP           ; UNARY ROUTINE & RETURN.

A03B  B180                  :EX010  LDA     (INLN),Y        ; RESTORE CHAR.
A03D  C928                          CMP     #'('            ; LEFT PAREN?
A03F  D00C ^A04D                    BNE     :EX020          ; NO.

A041  C8                            INY
A042  20D99F                        JSR     EXPRC           ; YES -- EVALUATE SUB-EXPRESSION.
A045  B180                          LDA     (INLN),Y
A047  C929                          CMP     #')'            ; MATCHING RIGHT PAREN?
A049  D0BF ^A00A                    BNE     EXP192          ; NO -- ERROR.

A04B  C8                            INY                     ; YES -- SKIP OVER IT.
A04C  60                            RTS

A04D  C93F                  :EX020  CMP     #'?'            ; RANDOM NUMBER?
A04F  D00D ^A05E                    BNE     :EX030          ; NO.

A051  AD0AD2                        LDA     PKYRND          ; YES -- GET RANDOM # FROM POKEY.
A054  8588                          STA     NUMBER
A056  AD0AD2                        LDA     PKYRND
A059  8589                          STA     NUMBER+1
A05B  C8                            INY                     ; SKIP OVER '?'.
A05C  D009 ^A067                    BNE     :EX040          ; (BRA).
```

```
                        ATARI CAMAC Assembler Ver  1.0A  Page 199
PILOT -- H.B. STEWART                                          D1:PILOT.

A05E  20xxxx        :EX030  JSR     ATOK
A061  D0A9 ^A00C            BNE     EXP194            ; ERROR.

A063  2986                  AND     #NUM+NVAR+BPTR    ; NUMERIC VARIABLE, POINTER OR CONSTANT?
A065  F0A3 ^A00A            BEQ     EXP192            ; NO -- ERROR.

A067  AE4E05        :EX040  LDX     ESTKP             ; RESULT TO STACK.
A06A  E0xE                  CPX     #ESTKSZ
A06C  F09C ^A00A            BEQ     EXP192            ; STACK OVERFLOW.

A06E  A58E                  LDA     NUMBER
A070  9593                  STA     EXPSTK,X
A072  A58F                  LDA     NUMBER+1
A074  9594                  STA     EXPSTK+1,X
A076  E8                    INX
A077  E8                    INX
A078  8E4E05                STX     ESTKP
A07B  60                    RTS


A07C                        PROC
                    ;
                    ; SOP -- STACK OPERATE
                    ;
                    ; CALLING SEQUENCE:
                    ;
A07C  A592          SOP     LDA     EXEC              ; EXECUTE?
A07E  F01D ^A09D            BEQ     :S0050            ; NO -- JUST REJUSTIFY THE STACK.

A080  84A7                  STY     TEMP2
A082  AD4E05                LDA     ESTKP             ; GET EXP STACK INDEX.
        = 0000              IF     DEBUG
   -                        CMP     #6                ; SEE IF STACK HAS AT LEAST 3 ENTRIES.
   -                        BCC     :S0090            ; NO -- PROBLEM!
                            ENDIF

A085  18                    CLC                       ; YES -- CONVERT STACK INDEX TO 'DTAB' INDEX.
A086  6911                  ADC     #EXPSTK-DTAB-2
A088  A8                    TAY
A089  AA                    TAX

A08A  CA                    DEX                       ; INDEX TO OPERATOR PROCESSOR ADDRESS.
A08B  CA                    DEX
A08C  B580                  LDA     DTAB,X            ; GET OPERATE ROUTINE ADDRESS.
A08E  8D0E05                STA     SJUMP+1
A091  B581                  LDA     DTAB+1,X
A093  8D0F05                STA     SJUMP+2

A096  CA                    DEX                       ; INDEX TO TARGET ENTRY.
A097  CA                    DEX
A098  200D05                JSR     SJUMP             ; OPERATE ON DATA.
A09B  A4A7                  LDY     TEMP2

A09D  38            :S0050  SEC                       ; (CLEAR BORROW).
A09E  AD4E05                LDA     ESTKP             ; ADJUST STACK INDEX.
A0A1  E904                  SBC     #4
A0A3  8D4E05                STA     ESTKP
```

```
A0A6  60                    RTS

      = 0000                IF      DEBUG
  -                :S0090   LDA      #INTERR          ; INTERNAL BUG.
  -                         JMP      PSTOP
                            ENDIF
```

```
A0A7                          PROC
                        ;
                        ; TEXP -- EVALUATE TEXT EXPRESSION
                        ;
                        ;       Y = POINTER TO START OF TEXT EXPR IN 'INLN'.
                        ;
                        ;       JSR     TEXP
                        ;       BNE     EXECUTE MODE
                        ;
                        ;       TEXP+2 = 0
                        ;       TEXP+3 = END OF TEXT EXPRESSION.
                        ;
                        ;       THE EOL IS NOT PART OF THE RESULTANT TEXT.
                        ;
A0A7  A592              TEXP    LDA     EXEC            ; EXECUTE MODE?
A0A9  D003 ^A0AE                BNE     :TE005          ; YES.

A0AB  4C1B9F                    JMP     SCNEOL          ; NO -- SCAN TO EOL & RETURN.

A0AE  A900              :TE005  LDA     #0              ; INIT RESULT LENGTH COUNT ...
A0B0  858F                      STA     TELN+3
A0B2  858E                      STA     TELN+2          ; ... & STARTING INDEX.

A0B4  AD3005                    LDA     CDEST           ; SAVE 'CHOT' DESTINATION.
A0B7  8D3105                    STA     CDEST+1
A0BA  A9FF                      LDA     #$FF            ; YES -- RE-ROUTE 'CHOT' OUTPUT TO ':TEBUF'.
A0BC  8D3005                    STA     CDEST

A0BF  B180              :TE010  LDA     (INLN),Y        ; GET A CHARACTER.
A0C1  20F99E                    JSR     CHKTRM          ; STATEMENT TERMINATOR?
A0C4  F05C ^A122                BEQ     :TE400          ; YES.

A0C6  C925                      CMP     #'%'             ; SPECIAL NUMBER?
A0C8  F013 ^A0DD                BEQ     :TE100          ; YES.

A0CA  C940                      CMP     #'@'             ; POINTER?
A0CC  F00F ^A0DD                BEQ     :TE100          ; YES.

A0CE  C923                      CMP     #'#'            ; NUMERIC VARIABLE DELIMITER?
A0D0  F00B ^A0DD                BEQ     :TE100          ; YES.

A0D2  C924                      CMP     #'$'            ; STRING VARIABLE DELIMITER?
A0D4  F007 ^A0DD                BEQ     :TE100          ; YES.

A0D6  C8                :TE020  INY
A0D7  208294                    JSR     CHOT            ; YES -- PRINT TEXT LITERAL.
A0DA  4CBFA0                    JMP     :TE010

A0DD  48                :TE100  PHA                     ; SAVE THE TEXT CHARACTER.
A0DE  98                        TYA                     ; SAVE THE Y REG.
A0DF  48                        PHA

A0E0  206E81                    JSR     ATOM            ; GET VALUE.
A0E3  F021 ^A106                BEQ     :TE220          ; O.K.

A0E5  68                :TE210  PLA                     ; NOT ATOM -- RESTORE Y REG ...
A0E6  A8                        TAY
```

```
A0E7  C8                      INY                          ; LOOK AHEAD.
A0E8  B180                    LDA      (INLN),Y            ; IS NEXT CHAR = DOUBLE QUOTE?
A0EA  C922                    CMP      #'"'
A0EC  D014 ^A102              BNE      :TF218              ; NO.

A0EE  68                      PLA                          ; YES -- FLUSH THE '%'.
A0EF  C8                      INY                          ; GET NEXT CHARACTER IN LITERAL.

A0F0  B180           :TE212   LDA      (INLN),Y
A0F2  20F99E                  JSR      CHKTRM              ; STATEMENT TERMINATOR?
A0F5  F02B ^A122              BEQ      :TE400              ; YES.

A0F7  C8                      INY
A0F8  C922                    CMP      #'"'                ; LITERAL TERMINATOR?
A0FA  F0C3 ^A0BF              BEQ      :TE010              ; YES -- BACK TO NORMAL SCAN

A0FC  208294                  JSR      CHOT                ; NOT PRINT LITERAL CHAR.
A0FF  4CF0A0                  JMP      :TE212

A102  88             :TE218   DEY                          ; SET INDEX BACK.
A103  68                      PLA                          ; ... & CHARACTER.
A104  D0D0 ^A0D6              BNE      :TE020              ; (BRA).

A106  C910           :TE220   CMP      #USVAR              ; UNDEFINED STRING?
A108  F0DB ^A0E5              BEQ      :TE210              ; YES -- PRINT LITERALLY.

A10A  C908                    CMP      #SVAR               ; DEFINED STRING?
A10C  F00A ^A118              BEQ      :TE300              ; YES -- PRINT VALUE.

                     ; NUMERIC DATA

A10E  68                      PLA                          ; NO -- MUST BE NUMERIC VALUE.
A10F  68                      PLA                          ; CLEAR STACK.

A110  A238                    LDX      #NUMBER-DTAB        ; VALUE OF NUMBER.
A112  20149E                  JSR      DECASC              ; CONVERT TO ASCII & OUTPUT.
A115  4CBFA0                  JMP      :TE010              ; CONTINUE.

                     ; STRING VARIABLE

A118  68             :TE300   PLA                          ; CLEAR THE STACK.
A119  68                      PLA

A11A  A242                    LDX      #OP-DTAB            ; INDEX TO STRING VALUE.
A11C  209797                  JSR      PRTSTG
A11F  4CBFA0                  JMP      :TE010

A122  AD3105         :TE400   LDA      CDEST+1             ; RESTORE 'CHOT' DESTINATION.
A125  8D3005                  STA      CDEST

                     ; *** EXTERNAL ENTRY POINT FROM 'XACCPT' ***

A128  A68F           TRAILB   LDX      TELN+3              ; EXAMINE LAST CHAR OF TEXP.
A12A  E48E                    CPX      TELN+2
A12C  F00C ^A13A              BEQ      :TE480              ; NULL RESULT.

A12E  BDFFBB                  LDA      TEXBUF-1,X          ; GET LAST CHAR IN BUFFER.
```

```
A131  C95F                CMP     #'_'            ; UNDERSCORE?
A133  D005 ^A13A          BNE     :TE480          ; NO.

A135  A920                LDA     #' '            ; YES -- REPLACE WITH BLANK.
A137  9DFFB6              STA     TEXBUF-1,X

A13A  A592        :TE480  LDA     EXEC            ; THE CC IS BEING SET TO REFLECT THE STATE
                                                  ; OF THE 'EXEC' FLAG BECAUSE EVERY SINGLE
                                                  ; JSR TO ':TEP' USED TO BE FOLLOWED BY A
                                                  ; 'LDA EXEC' INSTRUCTION.  THESE HAVE ALL BEEN
                                                  ;"COMMENTED" OUT; WHEN WILL THIS ALL END?
A13C  60                  RTS
```

```
                          ;
                          ; HEREIN RESIDE THE LOWER LEVEL GRAPHICS ROUTINES FOR PILOT GRAPHICS.
                          ;

A13D                              PROC
                          ;
                          ; GMODE -- GRAPHICS 'MODE' SUBCOMMAND.
                          ;

A13D  20C49F      GMODE   JSR     EXP                ; GET MODE #.

A140  A592                LDA     EXEC               ; EXECUTE MODE.
A142  F026 ^A16A          BEQ     :GM090             ; NO.

A144  84AB                STY     XTEMP
A146  A000                LDY     #0                 ; SEE IF MODE IS 0-15.
A148  A910                LDA     #16
A14A  A213                LDX     #EXPSTK-DTAB
A14C  200F9C              JSR     DCWCI
A14F  B01A ^A16B          BCS     :GM092             ; NO -- MODE >=16.

A151  A693                LDX     EXPSTK             ; SEE IF ALLOWED AS GRAPHICS MODE.
A153  BDF6B7              LDA     GCHAR,X            ; WILL BE ZERO IF NOT ALLOWED.
A156  F013 ^A16B          BEQ     :GM092             ; NOT AN ALLOWED MODE.

A158  AD5205              LDA     SPLTSC             ; SEE IF SPLIT DESIRED.
A15B  F005 ^A162          BEQ     :GM020             ; NO.

A15D  3DE6B7              AND     GCHAR,X            ; YES -- IS SPLIT ALLOWED?
A160  F010 ^A172          BEQ     :GM094             ; NO -- ERROR.

A162  8E3705      :GM020  STX     GSMODE             ; YES -- SAVE MODE.
A165  201095              JSR     GSOPEN             ; RE-OPEN GRAPHICS SCREEN.

A168  A4AB                LDY     XTEMP

A16A  60          :GM090  RTS

A16B  A922        :GM092  LDA     #MODERR            ; ILLEGAL GRAPHICS MODE.
A16D  A4AB                LDY     XTEMP
A16F  4C3A7A              JMP     PSTOP

A172              GSP094
A172  A921        :GM094  LDA     #SPTERR            ; SPLIT SCREEN NOT ALLOWED.
A174  A4AB                LDY     XTEMP
A176  4C3A7A              JMP     PSTOP

A179                      PROC
                          ;
                          ; GFULL -- GRAPHICS 'FULL' SUBCOMMAND.
                          ;

A179  A592        GFULL   LDA     EXEC               ; EXECUTE MODE?
A17B  F015 ^A192          BEQ     :GF090             ; NO.

A17D  A5FF                LDA     RUN                ; RUN MODE?
A17F  F012 ^A193          BEQ     :GF092             ; NO -- ERROR.
```

```
A181  AD4505              LDA     SGLSTF          ; SINGLE STOP?
A184  D00D ^A193          BNE     :GF092          ; YES -- ERROR.

A186  A900                LDA     #0              ; FULL SCREEN
A188  8D5205              STA     SPLTSC

A18B  84AB                STY     XTEMP
A18D  201095              JSR     GSOPEN          ; OPEN SCREEN.
A190  A4AB                LDY     XTEMP

A192  60         :GF090   RTS

A193  A983       :GF092   LDA     #NRCERR
A195  4C3A7A              JMP     PSTOP

A198                      PROC
                 ;
                 ; GSPLIT -- GRAPHICS 'SPLIT' SUBCOMMAND.
                 ;
A198  A592       GSPLIT   LDA     EXEC            ; EXECUTE MODE?
A19A  F014 ^A1B0          BEQ     :GS090          ; NO.

A19C  A910                LDA     #SPLIT          ; SPLIT SCREEN.
A19E  8D5205              STA     SPLTSC

A1A1  AE3705              LDX     GSMODE          ; SEE IF SPLIT ALLOWED.
A1A4  3DE6B7              AND     GCHAR,X
A1A7  F0C9 ^A172          BEQ     GSP094          ; NO -- ERROR.

A1A9  84AB                STY     XTEMP
A1AB  201095              JSR     GSOPEN          ; YES -- OPEN SCREEN.
A1AE  A4AB                LDY     XTEMP

A1B0  60         :GS090   RTS

A1B1                      PROC
                 ;
                 ; 'DRAWTO', 'FILLTO' & 'GOTO' SUB-COMMAND PROCESSORS.
                 ;
A1B1  A912       GFILTO   LDA     #FILLTO         ; PEN DOWN.
A1B3  D006 ^A1BB          BNE     :GG005          ; (BRA).

A1B5  A90A       GDRWTO   LDA     #DRAWTO         ; PEN DOWN.
A1B7  D002 ^A1BB          BNE     :GG005          ; (BRA).

A1B9  A906       GGOTO    LDA     #GOTO           ; PEN UP.

A1BB  8DD405     :GG005   STA     GROPR           ; SET PEN POSITION.
A1BE  20D49F              JSR     EXP             ; GET X-COORDINATE.

A1C1  A592                LDA     EXEC            ; EXECUTE MODE?
A1C3  F008 ^A1CD          BEQ     :GG010          ; NO.

A1C5  A593                LDA     EXPSTK          ; YES -- UPDATE X.
A1C7  85E6                STA     GXNEW
```

```
A1C9  A594                  LDA     EXPSTK+1
A1CB  85E7                  STA     GXNEW+1

A1CD  20079F       :GG010   JSR     SKPSEP          ; SKIP OPERAND SEPARATOR.
A1D0  20D49F                JSR     EXP             ; GET Y-COORDINATE.

A1D3  A592                  LDA     EXEC            ; EXECUTE MODE?
A1D5  F011 ^A1E8            BEQ     :GG090          ; NO.

A1D7  A593                  LDA     EXPSTK          ; YES -- UPDATE Y.
A1D9  85E9                  STA     GYNEW
A1DB  A594                  LDA     EXPSTK+1
A1DD  85EA                  STA     GYNEW+1

                   ; *** EXTERNAL ENTRY POINT FROM 'GHOME' ***

A1DF  A900         GGT030   LDA     #0              ; CLEAR FRACTIONAL PORTION OF X & Y.
A1E1  85E8                  STA     GXNEW+2
A1E3  85EB                  STA     GYNEW+2

A1E5  207AA6                JSR     GMOVE           ; NOW EFFECT MOVE.

A1E8               GG0090
A1E8               GTR090
A1E8               GTT090
A1E8  60           :GG090   RTS                     ; RETURN.



A1E9                        PROC
A1E9  20D49F       GTRNTO   JSR     EXP             ; GET POLAR ANGLE.

A1EC  A592                  LDA     EXEC            ; EXECUTE MODE?
A1EE  F0F8 ^A1E8            BEQ     GTT090          ; NO.

A1F0  A593                  LDA     EXPSTK          ; YES -- UPDATE POLAR ANGLE.
A1F2  85F2                  STA     THETA
A1F4  A594                  LDA     EXPSTK+1
A1F6  85F3                  STA     THETA+1

A1F8  4C964B                JMP     MOD360          ; MODULO 360 & RETURN.

A1FB                        PROC
A1FB  A909         GBK      LDA     #DRAW           ; BK N = FD-N.
A1FD  8D0405                STA     GROPR
A200  20D49F                JSR     EXP             ; GET MAGNITUDE OF MOVE.
A203  A213                  LDX     #EXPSTK-DTAB    ; NEGATE IT.
A205  20F19C                JSR     DNEGI
A208  4C1BA2                JMP     :GG010          ; GO TO COMMON CODE.

A20B  A911         GFIL     LDA     #FILL           ; PEN DOWN.
A20D  D006 ^A215            BNE     :GG005

A20F  A909         GDRW     LDA     #DRAW           ; PEN DOWN.
A211  D002 ^A215            BNE     :GG005

A213  A905         GGO      LDA     #GO             ; PEN UP.
```

```
A215  8D0405      :GG005  STA     GROPR           ; SET PEN POSITION.
A218  20D49F              JSR     EXP             ; GET MAGNITUDE OF MOVE.

A21B  A592        :GG010  LDA     EXEC            ; EXECUTE MODE?
A21D  F0C9 ^A1E8          BEQ     GG0090          ; NO.

A21F  2032A2              JSR     CALDEL          ; CALCULATE GXNEW & GYNEW.
A222  207AA6              JSR     GMOVE           ; NOW EFFECT MOVE.

A225  ADC505              LDA     RBTON           ; IS ROBOT TURTLE ON?
A228  F007 ^A231          BEQ     :GG090          ; NO.

A22A  84AB                STY     XTEMP           ; SAVE INDEX.
A22C  200783              JSR     RGO             ; MOVE ROBOT ALSO.
A22F  A4AB                LDY     XTEMP           ; RESTORE INDEX.

A231  60          :GG090  RTS

A232  A901        CALDEL  LDA     #1              ; COS(THETA) = SIN(THETA+90).
A234  203BAD              JSR     SINVAL          ; GYNEW = GYNEW + (<EXP> * COS(THETA)).
A237  20F6AD              JSR     TMULT
A23A  A269                LDX     #GYNEW-DTAB
A23C  2044AE              JSR     TADDI

A23F  A900                LDA     #0
A241  203BAD              JSR     SINVAL          ; GXNEW = GXNEW + (<EXP> * SIN(THETA)).
A244  20F6AD              JSR     TMULT
A247  A266                LDX     #GXNEW-DTAB
A249  4C44AE              JMP     TADDI

A24C                      PROC
A24C  20D49F      GLT     JSR     EXP             ; LT N = RT -N.
A24F  A213                LDX     #EXPSTK-DTAB
A251  20F19C              JSR     DNEGI
A254  4C5AA2              JMP     :GT010          ; GO TO COMMON CODE.

A257  20D49F      GTRN    JSR     EXP             ; POLAR ANGLE DELTA THETA.

A25A  A592        :GT010  LDA     EXEC            ; EXECUTE MODE?
A25C  F08A ^A1E8          BEQ     GTR090          ; NO.

A25E  84AB                STY     XTEMP           ; YES -- SAVE INDEX.
A260  A272                LDX     #THETA-DTAB     ; THETA = THETA + DELTA.
A262  A013                LDY     #EXPSTK-DTAB
A264  20329C              JSR     DADDI

A267  2096AB              JSR     MOD360          ; MODULO 360.

A26A  ADC505              LDA     RBTON           ; IS ROBOT TURTLE ON?
A26D  F003 ^A272          BEQ     :GT090          ; NO.
A26F  20F983              JSR     RTURN           ; MOVE ROBOT ALSO.

A272  A4AB        :GT090  LDY     XTEMP
A274  60                  RTS

A275                      PROC
```

```
                          ;
                          ; GPEN -- GRAPHICS 'PEN' SUBCOMMAND
                          ;

A275  2096A4    GPEN     JSR     CLRMAT         ; SEE IF COLOR MATCH.
A278  D02E ^A2A8         BNE     :GP099         ; NO -- ERROR.

A27A  8DB805             STA     PENCOL         ; SAVE COLOR REGISTER VALUE.
A27D  A592               LDA     EXEC           ; EXECUTE MODE?
A27F  F026 ^A2A7         BEQ     :GP090         ; NO.

A281  B010 ^A293         BCS     :GP040         ; YES -- JIF 'UP', 'DOWN' OR 'ERASE'.

A283  8A                 TXA                    ; IS COLOR ALREADY AVAILABLE?
A284  1008 ^A28E         BPL     :GP030         ; YES.

A286  ADB805             LDA     PENCOL         ; NO -- FIND VACANT SLOT FOR NEW COLOR.
A289  20DEA4             JSR     CASSGN
A28C  D01A ^A2A8         BNE     :GP099         ; NO FREE SLOTS.

A28E  8A        :GP030   TXA                    ; MERGE PEN UP/DOWN STATUS WITH ...
      = 0000             IF      FALSE
   -                     EOR     PEN            ; ... NEW PIXEL VALUE.
   -                     AND     #$7F
   -                     EOR     PEN
                         ENDIF
A28F  8D1305             STA     PEN
A292  60                 RTS

A293  8A        :GP040   TXA
A294  F0F8 ^A28E         BEQ     :GP030         ; 'ERASE'.

A296  1007 ^A29F         BPL     :GP050         ; 'DOWN'.

A298  0D1305             ORA     PEN            ; 'UP'.
A29B  8D1305             STA     PEN
A29E  60                 RTS

A29F  AD1305    :GP050   LDA     PEN            ; 'DOWN'.
A2A2  297F               AND     #$FF-PCUP
A2A4  8D1305             STA     PEN

A2A7  60        :GP090   RTS

A2A8  4C3A7A    :GP099   JMP     PSTCF

A2AB  A592      GPU      LDA     EXEC           ; PEN UP.
A2AD  F0F8 ^A2A7         BEQ     :GP090

A2AF  A280               LDX     #PCUP
A2B1  4C93A2             JMP     :GP040

A2B4  A592      GPD      LDA     EXEC           ; PEN DOWN.
A2B6  F0EF ^A2A7         BEQ     :GP090

A2B8  A240               LDX     #PCDN
A2BA  4C93A2             JMP     :GP040
```

```
A2BD  A592          GPE     LDA     EXEC                ; PEN ERASE.
A2BF  F0E6 ^A2A7            BEQ     :GP090

A2C1  A200                  LDX     #0
A2C3  4C93A2                JMP     :GP040

A2C6                        PROC
                    ;
                    ; GBACK -- GRAPHICS 'BACKGROUND' SUBCOMMAND
                    ;

A2C6  2096A4        GBACK   JSR     CLRMAT              ; SEE IF COLOR MATCH.
A2C9  D014 ^A2DF            BNE     :GB099              ; NO -- ERROR.

A2CB  B010 ^A2DD            BCS     :GB092              ; JIF 'UP', 'DOWN' OR 'ERASE'.

A2CD  8DB805                STA     PENCOL              ; YES -- SAVE COLOR VALUE.
A2D0  A592                  LDA     EXEC                ; EXECUTE MODE?
A2D2  F008 ^A2DC            BEQ     :GB090              ; NO.

A2D4  A200                  LDX     #0                  ; INDEX FOR BACKGROUND.
A2D6  ADB805                LDA     PENCOL              ; COLOR REGISTER VALUE.
A2D9  20F7A4                JSR     SETCLR              ; SET 'PNCLRS' AND COLOR REGISTER.

A2DC  60            :GB090  RTS

A2DD  A902          :GB092  LDA     #IMPERR             ; OPERAND ERROR.

A2DF  4C3A7A        :GB099  JMP     PSTOP


A2E2                        PROC
                    ;
                    ; GCHNGE -- GRAPHICS 'CHANGE' SUBCOMMAND
                    ;

A2E2  2096A4        GCHNGE  JSR     CLRMAT              ; GET "FROM" OPERAND.
A2E5  D030 ^A317            BNE     :GC099              ; ERROR.

A2E7  B023 ^A30C            BCS     :GC092              ; 'UP', 'DOWN' OR 'ERASE' INVALID.

A2E9  A592                  LDA     EXEC                ; EXECUTE MODE?
A2EB  F003 ^A2F0            BEQ     :GC020              ; NO.

A2ED  8A                    TXA                         ; SEE IF "FROM" COLOR EXISTS.
A2EE  301C ^A30C            BMI     :GC092              ; NO -- ERROR.

A2F0  8EB705        :GC020  STX     PENNUM              ; YES -- SAVE PEN NUMBER.

A2F3  20079F                JSR     SKPSEP
A2F6  2096A4                JSR     CLRMAT              ; GET "TO" COLOR OPERAND.
A2F9  D01C ^A317            BNE     :GC099              ; ERROR.

A2FB  B00F ^A30C            BCS     :GC092              ; 'UP', 'DOWN' OR 'ERASE' INVALID.

A2FD  E0FF                  CPX     #$FF                ; CHECK FOR DOUBLE ASSIGN AFTER CHG.
```

```
A2FF  D010 ^A311            BNE     :GC094          ; DOUBLE ASSIGN -- ERROR.

A301  A692                  LDX     EXEC            ; EXECUTE MODE?
A303  F006 ^A30B            BEQ     :GC090          ; NO.

A305  AEB705                LDX     PENNUM          ; GET PEN NUMBER.
A308  20F7A4                JSR     SETCLR          ; SET 'PNCLRS' AND COLOR REGISTER.

A30B  60           :GC090   RTS

A30C  A902         :GC092   LDA     #IMPERR         ; INVALID OPERAND.
A30E  4C3A7A                JMP     PSTOP

A311  A592         :GC094   LDA     EXEC            ; NO PROBLEM IF NOT EXECUTE.
A313  F0F6 ^A30B            BEQ     :GC090

A315  A926                  LDA     #DCAERR         ; DOUBLE ASSIGN.

A317  4C3A7A       :GC099   JMP     PSTOP

A31A                        PROC
                   ;
                   ; GSHADE -- GRAPHICS 'SHADE' SUBCOMMAND.
                   ;
A31A  2096A4       GSHADE   JSR     CLRMAT          ; MATCH OPERAND.
A31D  D031 ^A350            BNE     :GS099          ; NO MATCH.

A31F  B02A ^A34B            BCS     :GS092          ; 'UP', 'DOWN' OR 'ERASE'.

A321  8DB805                STA     PENCOL          ; SAVE PEN COLOR.
A324  A592                  LDA     EXEC            ; EXECUTE MODE?
A326  F022 ^A34A            BEQ     :GS090          ; NO.

A328  8A                    TXA
A329  1008 ^A333            BPL     :GS030          ; COLOR ASSIGNED.

A32B  ADB805                LDA     PENCOL
A32E  20DEA4                JSR     CASSGN          ; COLOR NOT ASSIGNED -- DO SO.
A331  D01D ^A350            BNE     :GS099          ; NO FREE SLOT.

A333  8E9B05       :GS030   STX     FCOLOR          ; SAVE FILL COLOR.

A336  205EAC                JSR     GREAD           ; CHECK FOR INBOUNDS.
A339  B00F ^A34A            BCS     :GS090          ; TURTLE OUT OF BOUNDS.

A33B  84AB                  STY     XTEMP
A33D  2098AF                JSR     FLOOD           ; SHADE THE AREA.
A340  A03E                  LDY     #GX1-DTAB       ; RESTORE VISIBLE TURTLE TO PROPER LOC.
A342  20FFAB                JSR     SETCUR
A345  209FAA                JSR     CNVRT
A348  A4AB                  LDY     XTEMP

A34A  60           :GS090   RTS

A34B  8A           :GS092   TXA
A34C  F0E5 ^A333            BEQ     :GS030          ; 'ERASE' OK.
```

```
A34E  A902                   LDA     #IMPERR

A350  4C3A7A        :GS099   JMP     PSTOP
A353                         PROC
                    ;
                    ; GWALL -- WALL SUBCOMMAND PROCESSOR.
                    ;

A353  A212          GWALL    LDX     #ALTABX          ; 'NONE'?
A355  20A67C                 JSR     SBCMAT
A358  D00D ^A367             BNE     :GW010           ; NO.

A35A  A592                   LDA     EXEC             ; EXECUTE MODE?
A35C  F008 ^A366             BEQ     :GW009           ; NO.

A35E  A900                   LDA     #0               ; YES -- CLEAR WALLS.
A360  8DCD05                 STA     WALLS
A363  8DCE05                 STA     WALLS+1

A366  60            :GW009   RTS

A367  2096A4        :GW010   JSR     CLRMAT           ; PEN/COLOR SELECTION?
A36A  D020 ^A38C             BNE     :GW092           ; NO.

A36C  B01E ^A38C             BCS     :GW092           ; YES -- JIF 'UP', 'DOWN' OR 'ERASE'.

A36E  A592                   LDA     EXEC             ; EXECUTE MODE?
A370  F019 ^A38B             BEQ     :GW090           ; NO.

A372  8A                     TXA
A373  3017 ^A38C             BMI     :GW092           ; COLOR NOT ASSIGNED TO A PEN.

A375  F015 ^A38C             BEQ     :GW092           ; BACKGROUND CAN'T BE A WALL.

A377  0A                     ASL     A
A378  AA                     TAX
A379  BD3EAC                 LDA     WMASK,X
A37C  0DCD05                 ORA     WALLS
A37F  8DCD05                 STA     WALLS
A382  BD3FAC                 LDA     WMASK+1,X
A385  0DCE05                 ORA     WALLS+1
A388  8DCE05                 STA     WALLS+1

A38B  60            :GW090   RTS

A38C  A902          :GW092   LDA     #IMPERR
A38E  4C3A7A                 JMP     PSTOP


A391                         PROC
                    ;
                    ; GEXIT -- GRAPHICS 'QUIT' SUBCOMMAND.
                    ;

A391  A592          GEXIT    LDA     EXEC             ; EXECUTE MODE?
A393  F00A ^A39F             BEQ     :GE090           ; NO.
```

```
A395  84AB                    STY     XTEMP
A397  207483                  JSR     RBTOFF              ; 'ROBOT TURTLE' OFF.
A39A  20F494                  JSR     TXOPEN              ; OPEN TEXT MODE SCREEN.
A39D  A4AB                    LDY     XTEMP

A39F                  GHM090
A39F                  GCL090
A39F  60              :GE090  RTS

A3A0                          PROC
                      ;
                      ; GCLEAR -- GRAPHICS 'CLEAR' SUBCOMMAND.
                      ;
                      ; *** CALLED BY 'XRUN' TOO ***

A3A0  A592            GCLEAR  LDA     EXEC                ; EXECUTE MODE?
A3A2  F0FB ^A39F              BEQ     GCL090              ; NO.

A3A4  A900                    LDA     #0                  ; TO AVOID ERROR $8D IF CURSOR AT LOWER ...
A3A6  8554                    STA     ROWCRS              ; ... RIGHT CORNER OF SCREEN.

A3A8  A97D                    LDA     #CLEAR              ; YES -- CLEAR GRAPHICS SCREEN ...
A3AA  4C8097                  JMP     TOUT                ; ... & RETURN.

A3AD                          PROC
                      ;
                      ; GCLRPN -- GRAPHICS 'CLEARPENS'.
                      ;
A3AD  A592            GCLRPN  LDA     EXEC                ; EXECUTE MODE?
A3AF  F0EE ^A39F              BEQ     GCL090              ; NO.

A3B1  A901                    LDA     #1                  ; YES -- CLEAR PEN SELECTS.
A3B3  8D8A05                  STA     NXTCLR
A3B6  60                      RTS


A3B7                          PROC
                      ; GHOME -- TURTLE HOME

A3B7  A592            GHOME   LDA     EXEC                ; EXECUTE MODE?
A3B9  F0E4 ^A39F              BEQ     GHM090              ; NO.

A3BB  A900                    LDA     #0                  ; YES -- SET TURTLE X & Y TO ZERO.
A3BD  85E6                    STA     GXNEW
A3BF  85E7                    STA     GXNEW+1
A3C1  85E9                    STA     GYNEW
A3C3  85EA                    STA     GYNEW+1
A3C5  A906                    LDA     #GOTO
A3C7  8DD405                  STA     GROPR               ; GOTO TYPE.
A3CA  4CDFA1                  JMP     GGT030

A3CD                          PROC
                      ; GNORTH -- TURTLE NORTH

A3CD  A592            GNORTH  LDA     EXEC                ; EXECUTE MODE?
A3CF  F009 ^A3DA              BEQ     :GN090              ; NO,.
```

```
A3D1  A900                 LDA    #0              ; YES -- SET THETA TO ZERO.
A3D3  85F2                 STA    THETA
A3D5  85F3                 STA    THETA+1
A3D7  2096AB               JSR    MOD360

A3DA  60          :GN090   RTS

A3DB                       PROC
                  ;
                  ; GEDGE -- 'EDGE' SUBCOMMAND
                  ;
A3DB  A20E        GEDGE    LDX    #EDTABX         ; 'FREE', 'HALT', WRAP', OR 'BOUNCE'.
A3DD  20A87C               JSR    SBCMAT
A3E0  D016 ^A3F8           BNE    :GE099          ; NO MATCH.

A3E2  A592                 LDA    EXEC            ; EXECUTE MODE?
A3E4  F011 ^A3F7           BEQ    :GE090          ; NO.

A3E6  8E5E05               STX    EDGRUL          ; YES -- SET RULE SELECT.
A3E9  E006                 CPX    #EFREE          ; RULE = FREE?
A3EB  F00A ^A3F7           BEQ    :GE090          ; YES.

A3ED  A26C                 LDX    #GX-DTAB        ; TURTLE IN SCREEN BOUNDS?
A3EF  200DAB               JSR    INTEST
A3F2  F003 ^A3F7           BEQ    :GE090          ; YES.

A3F4  2087A3               JSR    GHOME           ; NO -- SEND TURTLE HOME.

A3F7  60          :GE090   RTS

A3F8  4C347A      :GE099   JMP    PSTOP

A3FB                       PROC
                  ; GTURTL -- TURTLE ON/OFF

A3FB  A20A        GTURTL   LDX    #ONOFFX         ; EXPECT 'ON' OR 'OFF'
A3FD  20A87C               JSR    SBCMAT
A400  D014 ^A416           BNE    :GT100          ; NO MATCH - SEE IF COLOR.

A402  A592                 LDA    EXEC            ; EXECUTE MODE?
A404  F00A ^A410           BEQ    :GT090          ; NO.

A406  8E4F05      :GT020   STX    TRTLON          ; YES -- SET TURTLE FLAG.
A409  84AB                 STY    XTEMP           ; SAVE INDEX.
A40B  200CA6               JSR    TRONOF          ; DEAL WITH TURTLE REP.
A40E  A4AB                 LDY    XTEMP           ; RESTORE INDEX.

A410  60          :GT090   RTS

A411  A902        :GT092   LDA    #IMPERR

A413  4C347A      :GT099   JMP    PSTOP

A416  2096A4      :GT100   JSR    CLRMAT          ; COLOR SELECTION?
A419  D0F8 ^A413           BNE    :GT099          ; NO -- ERROR.

A41B  B0F4 ^A411           BCS    :GT092          ; JIF 'UP', 'DOWN', OR 'ERASE'.
```

```
A41D  A692                LDX      EXEC             ; EXECUTE MODE?
A41F  F0EF ^A410          BEQ      :GT090           ; NO.

A421  8DC405              STA      TRTCOL           ; YES -- UPDATE TURTLE COLOR.
A424  D0E0 ^A406          BNE      :GT020           ; (BRA) WITH X <> 0.


A426                      PROC
                  ;
                  ; XSETP -- 'SETPEN' COMMAND PROCESSOR
                  ; XSETL -- 'SETLET' COMMAND PROCESSOR
                  ;

A426  F012 ^A43A  XSETL   BEQ      :SP010           ; SYNTAX SCAN ONLY.

A428  20BB96              JSR      TSTMOD           ; LETTERS MEDIUM OR LARGE?
A42B  C902                CMP      #TXML
A42D  D064 ^A493          BNE      :SP094           ; NO -- ERROR.

A42F  F009 ^A43A          BEQ      :SP010           ; (BRA) TO COMMON CODE.


A431  F007 ^A43A  XSETP   BEQ      :SP010           ; SYNTAX SCAN ONLY.

A433  20BB96              JSR      TSTMOD           ; GRAPHICS MODE?
A436  290C                AND      #GRSS+GRFS
A438  F059 ^A493          BEQ      :SP094           ; NO -- ERROR.

A43A  20BDA4      :SP010  JSR      CLM040           ; GET PEN NUMBER.
A43D  D050 ^A48F          BNE      :SP090           ; ERROR.

A43F  8EB705              STX      PENNUM           ; SAVE PEN NUMBER.

A442  20079F              JSR      SKPSEP           ; SEE IF ALPHA -- IF SO CALL CLRMAT, ETC.
A445  B180                LDA      (INLN),Y
A447  20919E              JSR      CLETTR           ; ALPHA CHARACTER?
A44A  B00D ^A459          BCS      :SP012           ; NO -- SHOULD BE NEXP.

A44C  2096A4              JSR      CLRMAT           ; YES -- SEE IF COLOR NAME?
A44F  D03E ^A48F          BNE      :SP090           ; NO -- ERROR.
A451  B03D ^A490          BCS      :SP092           ; NO -- 'UP', 'DOWN' OR 'ERASE'.

A453  A692                LDX      EXEC             ; EXECUTE MODE?
A455  F038 ^A48F          BEQ      :SP090           ; NO.

A457  902E ^A487          BCC      :SP030           ; VALID COLOR.

A459  20D49F      :SP012  JSR      EXP              ; GET HUE VALUE.
A45C  A592                LDA      EXEC             ; EXECUTE MODE?
A45E  F010 ^A470          BEQ      :SP020           ; NO.

A460  A594                LDA      EXPSTK+1
A462  D02C ^A490          BNE      :SP092           ; OUT OF RANGE.

A464  A593                LDA      EXPSTK
```

```
A466  C910                      CMP     #$10
A468  B026 ^A490                BCS     :SP092          ; OUT OF RANGE.

A46A  0A                        ASL     A               ; JUSTIFY THE HUE VALUE.
A46B  0A                        ASL     A
A46C  0A                        ASL     A
A46D  0A                        ASL     A
A46E  85AB                      STA     XTEMP           ; SAVE ADJUSTED VALUE.

A470  20079F           :SP020   JSR     SKPSEP
A473  20049F                    JSR     EXP             ; GET LUM VALUE.
A476  A592                      LDA     EXEC            ; EXECUTE MODE?
A478  F015 ^A48F                BEQ     :SP090          ; NO.

A47A  A594                      LDA     EXPSTK+1
A47C  D012 ^A490                BNE     :SP092          ; OUT OF RANGE.

A47E  A593                      LDA     EXPSTK
A480  C908                      CMP     #8
A482  B00C ^A490                BCS     :SP092          ; OUT OF RANGE.

A484  0A                        ASL     A               ; X2.
A485  05AB                      ORA     XTEMP           ; MERGE HUE WITH LUM.

A487  AE8705           :SP030   LDX     PENNUM          ; GET PEN NUMBER.
A48A  20F7A4                    JSR     SETCLR          ; SET 'PNCLRS' AND COLOR REGISTER.
A48D  A900                      LDA     #0              ; SET CC FOR NORMAL RETURN.

A48F  60               :SP090   RTS                     ; RETURN WITH CC SET.

A490  A902             :SP092   LDA     #IMPERR
A492  60                        RTS

A493  A983             :SP094   LDA     #NRCERR
A495  60                        RTS

A496                            PROC
                       ;
                       ; CLRMAT -- COLOR MATCHER
                       ;
                       ; CALLING SEQUENCE:
                       ;
                       ;        'INLN' = POINTER TO STATEMENT.
                       ;        Y = STATEMENT INDEX.
                       ;
                       ;        JSR     CLRMAT
                       ;        BNE     ERROR
                       ;
                       ;        C = 1 INDICATES X = 'PCUP', 'PCDN' OR 0.
                       ;        C = 0 INDICATES A = COLOR REGISTER VALUE.
                       ;                        X = -1 IF NOT IN 'PNCLRS', OR
                       ;                        X = PIXEL VALUE ('PNCLRS' SLOT #).

A496  A206             CLRMAT   LDX     #PCTABX         ; MATCH OPERAND.
A498  20AB7C                    JSR     SECMAT
A49B  D020 ^A4BD                BNE     :CM040          ; NO MATCH -- SEE IF NEXP.
```

```
A49D  E080                    CPX    #PCUP         ; CHECK FOR 'UP', 'DOWN', OR 'ERASE'.
A49F  F038 ^A4D9              BEQ    :CM080        ; 'UP'.

A4A1  E040                    CPX    #PCDN         ; 'DOWN'.
A4A3  F034 ^A4D9              BEQ    :CM080

A4A5  8A                      TXA
A4A6  F031 ^A4D9              BEQ    :CM080        ; 'ERASE'.

A4A8  A200                    LDX    #0            ; SEARCH 'PNCLRS' FOR VALUE MATCH.

A4AA  E8             :CM010   INX
A4AB  EC6A05                  CPX    NXTCLR
A4AE  B007 ^A4B7              BCS    :CM020        ; END OF VALID ENTRIES.

A4B0  DD8805                  CMP    PNCLRS,X      ; COLOR VALUE MATCH?
A4B3  D0F5 ^A4AA              BNE    :CM010        ; NO.

A4B5  18                      CLC                  ; YES -- INDICATE COLOR VALUE O.K.
A4B6  60                      RTS                  ; RETURN WITH CC SET.

A4B7  A2FF           :CM020   LDX    #$FF          ; INDICATE NOT FOUND.
A4B9  E0FF                    CPX    #$FF          ; SET CC.
A4BB  18                      CLC                  ; INDICATE COLOR VALUE O.K.
A4BC  60                      RTS                  ; RETURN WITH CC SET.

                     ; *** EXTERNAL ENTRY POINT FROM 'XSETP' & 'XSETL' ***

A4BD                 :CM040
A4BD  20049F         CLM040   JSR    EXP           ; PROCESS AS A NUMERIC EXPRESSION.
A4C0  A592                    LDA    EXEC          ; EXECUTE MODE?
A4C2  F00D ^A4D1              BEQ    :CM050        ; NO.

A4C4  A694                    LDX    EXPSTK+1
A4C6  D013 ^A4DB              BNE    :CM092        ; OUT OF RANGE.

A4C8  A693                    LDX    EXPSTK
A4CA  EC8905                  CPX    NCOLRS        ; IS VALUE IN RANGE?
A4CD  F002 ^A4D1              BEQ    :CM050        ; YES.
A4CF  B00A ^A4DB              BCS    :CM092        ; NO.

A4D1  BD8805         :CM050   LDA    PNCLRS,X      ; YES -- GET COLOR VALUE.
A4D4  DD8805                  CMP    PNCLRS,X      ; SET CC FOR EXIT.
A4D7  18                      CLC                  ; INDICATE PEN NUMBER O.K.
A4D8  60                      RTS                  ; RETURN WITH CC SET.

A4D9  38             :CM080   SEC                  ; X = 'PCUP' OR 'PCDN' OR 0.
A4DA  60                      RTS                  ; RETURN WITH CC SET.

A4DB  A902           :CM092   LDA    #IMPERR       ; OUT OF RANGE PEN NUMBER.
A4DD  60                      RTS

A4DE                          PROC
                     ;
                     ; CASSGN -- COLOR ASSIGNMENT
                     ;
                     ; CALLING SEQUENCE:
                        ;
                        ;       A = COLOR REGISTER VALUE
                        ;       GSMODE = GRAPHICS MODE
                        ;       NXTCLR = NEXT AVAILABLE SLOT NUMBER
                        ;       NCOLRS = LAST SLOT NUMBER
                        ;
                        ;       JSR     CASSGN
                        ;       BNE     ERROR
                        ;
                        ;       X = PEN NUMBER
                        ;

A4DE  AEBA05     CASSGN LDX     NXTCLR          ; GET NEXT SLOT NUMBER.
A4E1  ECB905            CPX     NCOLRS          ; USEABLE SLOT?
A4E4  F002 ^A4E8        BEQ     :CN005          ; YES.
A4E6  B00C ^A4F4        BCS     :CN092          ; NO MORE SLOTS.

A4E8  20F7A4     :CN005 JSR     SETCLR          ; ASSIGN COLOR TO PEN & COLOR REG.
A4EB  AEBA05            LDX     NXTCLR
A4EE  EEBA05            INC     NXTCLR

A4F1  A900              LDA     #0              ; SET CC FOR NORMAL EXIT.
A4F3  60                RTS

A4F4  A925       :CN092 LDA     #NMCERR         ; NO MORE PEN SLOTS.
A4F6  60                RTS

A4F7                    PROC
                        ;
                        ; SETCLR -- SET COLOR
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       A = COLOR REGISTER VALUE.
                        ;       X = PEN NUMBER (PIXEL VALUE).
                        ;       GSMODE = GRAHICS MODE.
                        ;
                        ;       JSR     SETCLR
                        ;

A4F7  9DB605     SETCLR STA     PNCLRS,X        ; FIRST SET PIXEL VAL IN TABLE.
A4FA  8CCB05            STY     SCTEMP
A4FD  48                PHA                     ; SAVE COLOR VALUE.
A4FE  8A                TXA                     ; PIXEL VALUE TO Y REGISTER.
A4FF  A8                TAY
A500  AD3705            LDA     GSMODE
A503  0A                ASL     A               ; X2.
A504  AA                TAX
A505  BD608A            LDA     COLADR,X        ; GET POINTER TO REGISTER SET.
A508  85F4              STA     FSTACK
A50A  BD618A            LDA     COLADR+1,X
A50D  85F5              STA     FSTACK+1
A50F  B1F4              LDA     (FSTACK),Y      ; GET COLOR REGISTER INDEX.
A511  AA                TAX
A512  68                PLA
A513  9DC002            STA     PCOLR0,X        ; STORE COLOR VALUE TO REGISTER.
A516  ACCB05            LDY     SCTEMP
```

```
A519  60                    RTS

A51A                        PROC
                    ;
                    ; PRCLNM -- FIND AND PRINT COLOR NAME
                    ;
                    ; CALLING SEQUENCE:
                    ;
                    ;       X = INDEX TO 'PNCLRS'.
                    ;
                    ;       JSR     PRCLNM
                    ;
A51A  BDFB05        PRCLNM  LDA     PNCLRS,X        ; GET COLOR REGISTER VALUE.
A51D  85A9                  STA     TEMP2+2
A51F  8ECC05                STX     PRTEMP          ; SAVE X REGISTER.
A522  A2FF                  LDX     #-1             ; SETUP TO SCAN THE NAME TABLE.

A524  E8            :PC010  INX
A525  86A8                  STX     TEMP2+1         ; SAVE INDEX TO START OF NAME.

A527  BD707F        :PC015  LDA     PCTAB,X         ; GET A CHARACTER.
A52A  F016 ^A542            BEQ     :PC080          ; END OF TABLE -- NO MATCH.

A52C  3003 ^A531            BMI     :PC020          ; FOUND THE 'SB' BYTE.

A52E  E8                    INX                     ; STILL INSIDE THE NAME.
A52F  D0F6 ^A527            BNE     :PC015          ; (BRA).

A531  E8            :PC020  INX                     ; BUMP TO THE VALUE BYTE.
A532  BD707F                LDA     PCTAB,X         ; GET THE VALUE.
A535  C5A9                  CMP     TEMP2+2         ; IS THIS THE ONE WE ARE LOOKING FOR?
A537  D0EB ^A524            BNE     :PC010          ; NO.

A539  A6A8                  LDX     TEMP2+1         ; YES -- GET INDEX TO NAME.
A53B  204FA5                JSR     PRNTCL          ; PRINT COLOR NAME.

A53E  AECC05                LDX     PRTEMP          ; RESTORE X REGISTER.
A541  60                    RTS

A542  A900          :PC080  LDA     #0              ; NO NAME -- PRINT THE NUMERIC VALUE.
A544  85AA                  STA     TEMP2+3         ; ZERO THE MSB FIRST.
A546  A229                  LDX     #TEMP2+2-DTAB   ; POINT TO NUMBER.
A548  20149E                JSR     DECASC

A54B  AECC05                LDX     PRTEMP          ; RESTORE X REGISTER.
A54E  60                    RTS

A54F                        PROC
                    ;
                    ; PRNTCL -- PRINT COLOR NAME FROM NAME TABLE.
                    ;
                    ; CALLING SEQUENCE:
                    ;
                    ;       X = INDEX TO FIRST CHARACTER OF COLOR NAME.
                    ;
                    ;       JSR     PRNTCL
                    ;
```

```
                        ;       X = INDEX TO NAME DELIMITER.
                        ;
A54F  BD707F       PRNTCL  LDA     PCTAB,X          ; GET A CHARACTER.
A552  3006 ^A55A           BMI     :PC090           ; DELIMITER.

A554  208294               JSR     CHOT
A557  E8                   INX
A558  D0F5 ^A54F           BNE     PRNTCL           ; (BRA).

A55A  60           :PC090  RTS
```

```
A55B                        PROC
A55B  A26C          TRTPLC  LDX     #GX-DTAB        ; TURTLE IN BOUNDS?
A55D  2000AB                JSR     INTEST
A560  F008 ^A56A            BEQ     :TR090          ; YES.

A562  A9FF                  LDA     #-1             ; NO -- SET FLAG.
A564  8D0005                STA     GCOL+1
A567  206BA5                JSR     CLRTRT          ; CLEAR OLD TURTLE.

A56A  60            :TR090  RTS

A56B  AE5F05        CLRTRT  LDX     TRYPOS          ; GET OLD POSITION.
A56E  A00E                  LDY     #VTHITE
A570  A900                  LDA     #0

A572  9D0577        :TP020  STA     TRBUFF,X        ; REMOVE OLD REPRESENTATION.
A575  E8                    INX
A576  88                    DEY
A577  D0F9 ^A572            BNE     :TP020

A579  60                    RTS


A57A                        PROC
                    ;
                    ; TRTLOC -- PLACE VISIBLE TURTLE (AT NEW LOC).
                    ;
                    ; CALLING SEQUENCE:
                    ;
                    ;       'TUFLAG'        =  0 IF GCOL & GROW O.K.
                    ;       'TRTLON'        =  0 IF OFF, ELSE ON.
                    ;       'GSMODE'        =  GRAPHICS SCREEN MODE.
                    ;       'THETA'         =  TURTLE ANGLE.
                    ;       'GCOL'          =  TURTLE X POSITION.
                    ;       'GROW'          =  TURTLE Y POSITION.
                    ;
                    ;       JSR     TRTLOC
                    ;
A57A  AD4F05        TRTLOC  LDA     TRTLON          ; TURTLE ON?
A57D  F061 ^A5E0            BEQ     :TP100          ; NO.

A57F  AD0905                LDA     TUFLAG          ; ARE PARMS VALID?
A582  D05C ^A5E0            BNE     :TP100          ; NOT NECESSARILY.

A584  AD0005                LDA     GCOL+1          ; IN SCREEN BOUND?
A587  3057 ^A5E0            BMI     :TP100          ; NO.

A589  206BA5                JSR     CLRTRT          ; CLEAR OLD TURTLE.
A58C  2039A6                JSR     DOMCAL          ; CALCULATE ORIENTATION.

                    ; CONVERT CURSOR X TO COLOR CLOCKS.

A58F  AE3705                LDX     GSMODE          ; SCREEN MODE DEPENDENT
A592  BC9CB8                LDY     CCPXTB,X        ; GET # OF COLOR CLOCKS PER X UNIT.
A595  F00C ^A5A3            BEQ     :TP040          ; ZERO INDICATES 1/2 CLOCK.
```

```
A597  98                   TYA                      ; START WITH 1/2 POSITION OFFSET.
A598  18                   CLC
A599  6A                   ROR     A
A59A  18                   CLC

A59B  6DCF05       :TP030  ADC     GCOL             ; NOW DO MULTIPLY.
A59E  88                   DEY
A59F  D0FA ^A59B           BNE     :TP030

A5A1  F008 ^A5AB           BEQ     :TP050           ; (BRA).

A5A3  ADD005       :TP040  LDA     GCOL+1           ; DIVIDE BY 2 (1/2 COLOR CLOCK).
A5A6  6A                   ROR     A
A5A7  ADCF05               LDA     GCOL
A5AA  6A                   ROR     A

A5AB  18           :TP050  CLC
A5AC  6930                 ADC     #$30             ; LEFT EDGE OFFSET.
A5AE  AC6005               LDY     ORIENT           ; SUBTRACT ORIENTATION OFFSET.
A5B1  38                   SEC
A5B2  F9F8B8               SBC     TRDX,Y
A5B5  18                   CLC
A5B6  8D03D0               STA     HPOS0+3          ; RESULT IS PLAYER3 HORIZONTAL POSITION.

                   ; CONVERT CURSOR Y TO SCAN LINES

A5B9  BCACB8               LDY     SLPYTB,X         ; GET #SCAN LINES PERR Y UNIT.
A5BC  98                   TYA                      ; START WITH 1/2 POSITION OFFSET.
A5BD  18                   CLC
A5BE  6A                   ROR     A
A5BF  18                   CLC

A5C0  6DD105       :TP060  ADC     GROW             ; MULTIPLY.
A5C3  88                   DEY
A5C4  D0FA ^A5C0           BNE     :TP060

A5C6  6915                 ADC     #$15             ; *** MAGIC OFFSET ***
A5C8  AC6005               LDY     ORIENT           ; SUBTRACT ORIENTATION OFFSET.
A5CB  38                   SEC
A5CC  F9E0B8               SBC     TRDY,Y
A5CF  8D5F05               STA     TRYPOS           ; SAVE FOR NEXT TIME IN.
A5D2  AA                   TAX                      ; SETUP FOR THIS TIME.
A5D3  A000                 LDY     #0

A5D5  B1F8         :TP090  LDA     (TRADDR),Y       ; MOVE PATTERN TO MISSILE BUFFER.
A5D7  9D0577               STA     TRBUFF,X
A5DA  E8                   INX
A5DB  C8                   INY
A5DC  C00E                 CPY     #VTHITE
A5DE  D0F5 ^A5D5           BNE     :TP090

A5E0  60           :TP100  RTS

A5E1                       PROC
                   ; TRTINI -- VISIBLE TURTLE INITIALIZATION.
```

```
A5E1                TRTINI
A5E1  A2FD                  LDX    #253              ; CLEAR TURTLE REPRESENTATION BUFFER.
A5E3  A900                  LDA    #0
A5E5  8D5F05                STA    TRYPOS
A5E8  8DD905                STA    TUFLAG            ; INITIALIZE TURTLE LOC, INTERLOCK.
A5EB  8D0BD0                STA    SIZEP3            ; PLAYER SIZE.

A5EE  9D0277        :TI010  STA    TPBUFF+2,X
A5F1  CA                    DEX
A5F2  D0FA ^A5EE            BNE    :TI010

A5F4  A208                  LDX    #8                ; INITIALIZE PLAYER/MISSILE HARDWARE.

A5F6  9DFFCF        :TI020  STA    HPOS0-1,X         ; SET ALL HORIZONTAL POSITION TO ZERO.
A5F9  CA                    DEX
A5FA  D0FA ^A5F6            BNE    :TI020

A5FC  A901                  LDA    #1                ; SET PRIORITY.
A5FE  8D6F02                STA    GPRIOR
A601  A970                  LDA    # HIGH [TPBUFF-$700]     ; PLAYER/MISSILE BASE ADDRESS.
A603  8D07D4                STA    PMBASE
A606  A902                  LDA    #$02              ; DEFAULT TURTLE COLOR.
A608  8DC405                STA    TRTCOL

A60B  60                    RTS

A60C                        PROC
                    ; TRONOF -- MISSILE DMA ON/OFF.

A60C  AD4F05        TRONOF  LDA    TRTLON            ; TURTLE ON?
A60F  F017 ^A628            BEQ    :TF050            ; NO.

A611  ADC405                LDA    TRTCOL            ; YES -- SET PLAYER COLOR REG.
A614  8DC302                STA    PCOLP0+3

A617  A902                  LDA    #2
A619  8D1DD0                STA    GRACTL

A61C  AD2F02                LDA    DMACT             ; ENABLE PLAYER DMA (HIGH RESOLUTION MODE).
A61F  0918                  ORA    #$18
A621  8D2F02                STA    DMACT
A624  8D00D4                STA    DMACTL

A627  60                    RTS

A628  AD2F02        :TF050  LDA    DMACT             ; PLAYER DMA OFF.
A62B  29E7                  AND    #$E7
A62D  8D2F02                STA    DMACT

A630  A900                  LDA    #0
A632  8D1DD0                STA    GRACTL
A635  8D10D0                STA    GRAFP3

A638  60                    RTS
```

```
A639                          PROC
A639  AD0205        DUMCAL  LDA     GANGLE          ; TRADDR := GANGLE.
A63C  85F8                  STA     TRADDR
A63E  AD0305                LDA     GANGLE+1
A641  85F9                  STA     TRADDR+1
A643  A000                  LDY     #0

A645  A278                  LDX     #TRADDR-DTAB
A647  A9F8                  LDA     #-8
A649  20049D                JSR     DADDS
A64C  A5F9                  LDA     TRADDR+1
A64E  3010 ^A660            BMI     :DC020

A650  C8            :DC010  INY
A651  A9F1                  LDA     #-15
A653  20049D                JSR     DADDS
A656  A5F9                  LDA     TRADDR+1
A658  10F6 ^A650            BPL     :DC010

A65A  C018                  CPY     #24
A65C  9002 ^A660            BCC     :DC020

A65E  A000                  LDY     #0

A660  8C6005        :DC020  STY     ORIENT
A663  A9B9                  LDA     # HIGH VTURT    ; SETUP POINTER TO TURTLE REP.
A665  85F9                  STA     TRADDR+1
A667  A910                  LDA     # LOW VTURT
A669  85F8                  STA     TRADDR
A66B  C000                  CPY     #0
A66D  F00A ^A679            BEQ     :DC090

A66F  A278                  LDX     #TRADDR-DTAB

A671  A90E          :DC030  LDA     #VTHITE         ; CALCULATE OFFSET.
A673  20049D                JSR     DADDS
A676  88                    DEY
A677  D0F8 ^A671            BNE     :DC030

A679  60            :DC090  RTS


A67A                        PROC
                    ; LOWER LEVEL GRAPHICS UTILITIES
                    ;
                    ; CALLING SEQUENCE:
                    ;
                    ;       'GX' & 'GY' = START COORDINATES.
                    ;       'GXNEW' & 'GYNEW' = END COORDINATES.
                    ;
                    ;       JSR     GMOVE
                    ;
                    ;       'GX' = 'GXNEW' = END COORDINATES.
                    ;       'GY' = 'GYNEW' = END COORDINATES.
                    ;
A67A  84E0          GMOVE   STY     LEND            ; SAVE Y REGISTER.
```

```
A67C  A20C                LDX     #12              ; 4 VARIABLES OF 3 BYTES EACH.

A67E  B5E5       :GM010   LDA     GXNEW-1,X        ; MOVE COORDINATES TO WORKING VARIABLES.
A680  2A                  ROL     A                ; PREPARE TO ROUND.
A681  B5E3                LDA     GXNEW-3,X        ; GET MIDDLE BYTE.
A683  6900                ADC     #0               ; ADD MSB OF FRACTION.
A685  95BB                STA     GX1-3,X
A687  B5E4                LDA     GXNEW-2,X        ; GET MSB.
A689  6900                ADC     #0               ; CONTINUE ROUNDING.
A68B  95BC                STA     GX1-2,X
A68D  A900                LDA     #0               ; NOW CLEAR FRACTION.
A68F  95BD                STA     GX1-1,X
A691  CA                  DEX
A692  CA                  DEX
A693  CA                  DEX
A694  D0E8 ^A67E          BNE     :GM010

                 ; *S*    LDA     #0
A696  8DD805              STA     NOPLOT

A699  ADD405              LDA     GROPR            ; GOTO?
A69C  C906                CMP     #GOTO
A69E  F018 ^A6B8          BEQ     :GM10F           ; YES.

A6A0  2071AB              JSR     NEWDEL           ; CALCULATE SLOPE DELTAS.
A6A3  5003 ^A6A8          BVC     :GM005
A6A5  4C78A7              JMP     :GM041           ; OVERFLOW.

A6A8  ADD405     :GM005   LDA     GROPR
A6AB  C90A                CMP     #DRAWTO          ; 'DRAWTO'?
A6AD  F035 ^A6E4          BEQ     :GM012           ; YES.

A6AF  2910                AND     #$10             ; 'FILL' OR 'FILLTO'?
A6B1  F008 ^A6BB          BEQ     :GM011           ; NO -- 'DRAW' OR 'GO'.

A6B3  AD1305              LDA     PEN              ; PEN ERASE ON FILL?
A6B6  D003 ^A6BB          BNE     :GM011           ; NO.

A6B8  4CCEA7     :GM10F   JMP     :GM150

A6BB  AD5E05     :GM011   LDA     EDGRUL           ; FREE?
A6BE  C908                CMP     #EFREE
A6C0  F00A ^A6CC          BEQ     :GM11F           ; YES -- CLIP.

A6C2  A244                LDX     #GX2-DTAB        ; IS START POINT IN BOUNDS?
A6C4  200DAB              JSR     INTEST
A6C7  D01B ^A6E4          BNE     :GM012           ; NO -- CLIP.

A6C9  4CB4A7              JMP     :GM120           ; YES -- HALT, WRAP OR BOUNCE.

                 ; CHECK FOR LINE SEGMENT WITHIN SCREEN LIMITS

                 ; THE CLIPPING ALGORITHM USED HERE IS DESCRIBED IN SECTION 5-1 OF THE
                 ; SECOND EDITION OF "PRINCIPLES OF INTERACTIVE COMPUTER GRAPHICS" BY
                 ; NEWMAN & SPROULL.

A6CC  ADD405     :GM11F   LDA     GROPR            ; "GO"?
```

```
A6CF  C905                    CMP     #GO
A6D1  D011 ^A6E4              BNE     :GM012          ; NO.

A6D3  AD5E05                  LDA     EDGRUL          ; EDGE RULE = FREE?
A6D6  C908                    CMP     #EFREE
A6D8  D00A ^A6E4              BNE     :GM012          ; NO.

A6DA  A23E                    LDX     #GX1-DTAB       ; END POINT IN BOUNDS?
A6DC  200DAB                  JSR     INTEST
A6DF  F003 ^A6E4              BEQ     :GM012          ; YES.

A6E1  EED805                  INC     NOPLOT          ; NO -- DON'T PLOT END POINT.

A6E4  A23E           :GM012   LDX     #GX1-DTAB       ; TEST END POINT.
A6E6  200DAB                  JSR     INTEST
A6E9  8D4905                  STA     GNUMB           ; SAVE RESULT.

A6EC  A244                    LDX     #GX2-DTAB       ; TEST START POINT.
A6EE  200DAB                  JSR     INTEST
A6F1  8D4A05                  STA     GNUMB+1         ; SAVE RESULT.

A6F4  2D4905                  AND     GNUMB
A6F7  F003 ^A6FC              BEQ     :GM013          ; PART OF LINE MAY BE IN SCREEN.

A6F9  4CE0A7         :GMOVF   JMP     :GM157          ; NO PART OF LINE IS IN SCREEN.

A6FC  AD4905         :GM013   LDA     GNUMB
A6FF  0D4A05                  ORA     GNUMB+1
A702  D003 ^A707              BNE     :GM014          ; PART OF LINE IS OFF THE SCREEN.

A704  4CAFA7                  JMP     :GM110          ; ALL OF LINE IS IN SCREEN.

A707  A23E           :GM014   LDX     #GX1-DTAB       ; FIND AN INTERSECTION WITH AN EDGE.
A709  AD4905                  LDA     GNUMB           ; IS X1,Y1 OUTSIDE SCREEN?
A70C  D005 ^A713              BNE     :GM016          ; YES.

A70E  A244                    LDX     #GX2-DTAB       ; NO -- THEN X2,Y2 MUST BE.
A710  AD4A05                  LDA     GNUMB+1

A713  48             :GM016   PHA                     ; SAVE INTERSECT STATUS.
A714  2908                    AND     #ELEFT          ; LEFT EDGE INTERSECTION?
A716  F00C ^A724              BEQ     :GM020          ; NO.

A718  38                      SEC                     ; YES -- 'GACC' = LEFT EDGE X VALUE.
A719  A900                    LDA     #0              ; 'GACC' = -XC.
A71B  ED6105                  SBC     XC
A71E  85CE                    STA     GACC
A720  A9FF                    LDA     #-1
A722  D00D ^A731              BNE     :GM025          ; (BRA).

A724  68             :GM020   PLA                     ; GET STATUS.
A725  48                      PHA
A726  2904                    AND     #ERIGHT         ; RIGHT EDGE INTERSECTION?
A728  F036 ^A760              BEQ     :GM030          ; NO.

A72A  AD6105                  LDA     XC              ; YES -- 'GACC' = RIGHT EDGE X VALUE.
A72D  85CE                    STA     GACC            ; 'GACC' = XC.
```

```
A72F  A900                  LDA     #0

A731  85CF         :GM025   STA     GACC+1            ; EXTEND SIGN.
A733  A04E                  LDY     #GACC-DTAB        ; GX1 OR GX2 = 'GACC'.
A735  20459A                JSR     DMOVI

A738  8A                    TXA                       ; GY1 OR GY2 = (GACC-GX) * DELY / DELX + GY.
A739  48                    PHA
A73A  A24E                  LDX     #GACC-DTAB
A73C  A06C                  LDY     #GX-DTAB
A73E  2057AF                JSR     RSUBI

A741  A04C                  LDY     #DELY-DTAB
A743  205BAE                JSR     GMULT

A746  A04A                  LDY     #DELX-DTAB
A748  208EAE                JSR     GDIV
A74B  D02B ^A778            BNE     :GM041            ; OVERFLOW -- DON'T DRAW.

A74D  A06F                  LDY     #GY-DTAB
A74F  2050AF                JSR     RADDI

A752  68                    PLA
A753  AA                    TAX
A754  A5CE                  LDA     GACC
A756  9583                  STA     DTAB+3,X
A758  A5CF                  LDA     GACC+1
A75A  9584                  STA     DTAB+4,X
A75C  68                    PLA                       ; CLEAR STACK.
A75D  4CF4A6                JMP     :GM012            ; KEEP THIS UP UNTIL LINE SEGMENT IS CLIPPED.

A760  68           :GM030   PLA                       ; GET STATUS.
A761  48                    PHA
A762  2902                  AND     #EBOTOM           ; BOTTOM EDGE INTERSECTION?
A764  F00C ^A772            BEQ     :GM040            ; NO.

A766  38                    SEC                       ; YES -- 'GACC' = BOTTOM EDGE Y VALUE.
A767  A900                  LDA     #0                ; 'GACC' = -YC.
A769  ED6305                SBC     YC
A76C  85CE                  STA     GACC
A76E  A9FF                  LDA     #-1
A770  D010 ^A782            BNE     :GM045            ; (BRA).

A772               :GM040
A772  68                    PLA                       ; GET STATUS.
A773  48                    PHA
A774  2901                  AND     #ETOP             ; TOP EDGE INTERSECTION?
A776  D003 ^A77B            BNE     :GM042            ; YES.

A778  4CF0A7       :GM041   JMP     :GM157

A77B               :GM042
A77B  AD6305                LDA     YC                ; 'GACC' = TOP EDGE Y VALUE.
A77E  85CE                  STA     GACC              ; 'GACC' = YC.
A780  A900                  LDA     #0

A782  85CF         :GM045   STA     GACC+1            ; EXTEND SIGN.
```

```
A784  9584                 STA     DTAB+4,X        ; GY1 OR GY2 = 'GACC'.
A786  A5CE                 LDA     GACC
A788  9583                 STA     DTAB+3,X

A78A  8A                   TXA                     ; SAVE X REGISTER.
A78B  48                   PHA                     ; GX1 OR GX2 = (GACC - GY) * DELX / DELY + GX.
A78C  A24E                 LDX     #GACC-DTAB
A78E  A06F                 LDY     #GY-DTAB
A790  2097AF               JSR     RSUBI

A793  A04A                 LDY     #DELX-DTAB
A795  205BAE               JSR     GMULT

A798  A04C                 LDY     #DELY-DTAB
A79A  208EAE               JSR     GDIV
A79D  D0D9 ^A778           BNE     :GM041          ; OVERFLOW. -- DON'T DRAW.

A79F  A06C                 LDY     #GX-DTAB
A7A1  2050AF               JSR     RADDI

A7A4  68                   PLA
A7A5  AA                   TAX
A7A6  A04E                 LDY     #GACC-DTAB
A7A8  20459A               JSR     DMOVI
A7AB  68                   PLA                     ; CLEAR THE STACK.
A7AC  4CE4A6               JMP     :GM012          ; KEEP THIS UP UNTIL LINE SEGMENT IS CLIPPED.

A7AF  2071A8       :GM110  JSR     NEWDEL          ; CALCULATE SLOPE DELTAS FOR CLIPPED LINE.
A7B2  70C4 ^A778           BVS     :GM041          ; OVERFLOW.

A7B4  201CA8       :GM120  JSR     NEWDRW          ; DRAW LINE.

A7B7  ADD505               LDA     HITWLL          ; HIT WALL?
A7BA  D00C ^A7C8           BNE     :GM130          ; YES.

A7BC  ADD405               LDA     GROPR
A7BF  C90A                 CMP     #DRAWTO         ; 'DRAWTO'?
A7C1  F01D ^A7E0           BEQ     :GM157          ; YES

A7C3  ADD605               LDA     HITEDG          ; HIT EDGE?
A7C6  F018 ^A7E0           BEQ     :GM157          ; NO.

A7C8  20E9A7       :GM130  JSR     SETCR2          ; SET CURSOR COORDINATES.
A7CB  4CE3A7               JMP     :GM160          ; (BRA).

                   ; GOTO

A7CE  A23E         :GM150  LDX     #GX1-DTAB       ; CHECK FOR POINT IN SCREEN.
A7D0  200DAB               JSR     INTEST
A7D3  D00B ^A7E0           BNE     :GM157          ; NOT IN SCREEN -- DON'T PLOT.

A7D5  A03E                 LDY     #GX1-DTAB
A7D7  20FFAB               JSR     SETCUR          ; CONVERT TO HANDLER COORDINATES.
A7DA  209FAA               JSR     CNVRT
A7DD  2093AA               JSR     PLOT            ; PLOT POINT IF PEN DOWN.

A7E0  208CA8       :GM157  JSR     NEWCUR          ; ESTABLISH NEW CURSOR POSITION.
```

```
A7E3  2056A5      :GM160  JSR     TRTPLC          ; PLACE VISIBLE TURTLE.
A7E6  A4F0                LDY     LEND              ; RESTORE Y REGISTER.
A7E8  60                  RTS

A7E9  38          SETCR2  SEC                     ; GX := GCOL-XC.
A7EA  ADCF05              LDA     GCOL            ; GXNEW := SAME.
A7ED  ED6105              SBC     XC
A7F0  85EC                STA     GX
A7F2  85E6                STA     GXNEW
A7F4  ADD005              LDA     GCOL+1
A7F7  ED6205              SBC     XC+1
A7FA  85ED                STA     GX+1
A7FC  85E7                STA     GXNEW+1

A7FE  38                  SEC                     ; GY := YC-GROW
A7FF  AD6305              LDA     YC              ; GYNEW := SAME.
A802  EDD105              SBC     GROW
A805  85EF                STA     GY
A807  85E9                STA     GYNEW
A809  A900                LDA     #0
A80B  E900                SBC     #0
A80D  85F0                STA     GY+1
A80F  85EA                STA     GYNEW+1

A811  A900                LDA     #0              ; CLEAR FRACTION.
A813  85EE                STA     GX+2
A815  85F1                STA     GY+2
A817  85E8                STA     GXNEW+2
A819  85EB                STA     GYNEW+2
A81B  60                  RTS
```

```
A81C                        PROC
                  ;
                  ; NEWDRW -- LINE DRAW ROUTINE
                  ;
                  ; CALLING SEQUENCE:
                  ;
                  ;       GX2,GY2 = START COORDINATE.
                  ;       GX1,GY1 = END COORDINATE.
                  ;       DELX,DELY DEFINES SLOPE OF LINE.
                  ;       PEN     = PIXEL VALUE.
                  ;       GSMODE  = SCREEN MODE.
                  ;       EDGRUL  = EDGE RULE IN EFFECT.
                  ;       GROPR   = OPERATION.
                  ;
                  ;       JSR     NEWDRW
                  ;
                  ;       HITWLL  = 0 IF NO WALL HIT
                  ;       HITEDG  = 0 IF NO EDGE HIT
                  ;       GCOL    = COL OF LAST DRAWN PIXEL
                  ;       GROW    = ROW OF LAST DRAWN PIXEL
                  ;
A81C  A044        NEWDRW  LDY     #GX2-DTAB       ; ROWCRS := YC - GY2.
A81E  20FFAB              JSR     SETCUR          ; COLCRS := GX2 + XC.
A821  209FAA              JSR     CNVRT
A824  A900                LDA     #0
A826  8DD705              STA     HALTFG
A829  8DD505              STA     HITWLL
A82C  8DD605              STA     HITEDG
A82F  ADD405              LDA     GROPR
A832  C905                CMP     #GO             ; 'GO'?
A834  F003 ^A839          BEQ     :DR020          ; YES -- DON'T PLOT START POINT.

A836  2093AA              JSR     PLOT            ; NO -- DRAW START POINT.

A839  A213        :DR020  LDX     #DELTAR-DTAB    ; DELTAR := ABS(DELY).
A83B  A04C                LDY     #DELY-DTAB
A83D  20459A              JSR     DMOVI
A840  20FF9C              JSR     DABSI

A843  A215                LDX     #DELTAC-DTAB    ; DELTAC := ABS(DELX).
A845  A04A                LDY     #DELX-DTAB
A847  20459A              JSR     DMOVI
A84A  20FF9C              JSR     DABSI

A84D  A213                LDX     #DELTAR-DTAB    ; IF DELTAR>DELTAC
A84F  A015                LDY     #DELTAC-DTAB
A851  20159C              JSR     DCMPI
A854  9028 ^A87E          BCC     :DR050

A856  A21E                LDX     #ENDPT-DTAB     ; THEN BEGIN.
A858  A013                LDY     #DELTAR-DTAB    ; ENDPT := DELTAR.
A85A  20459A              JSR     DMOVI
A85D  A21A                LDX     #COLAC-DTAB     ; COLAC := DELTAR/2.
A85F  20459A              JSR     DMOVI
A862  4698                LSR     COLAC+1
A864  669A                ROR     COLAC
A866  A900                LDA     #0              ; ROWAC := 0.
```

```
A868  8598              STA   ROWAC
A86A  8599              STA   ROWAC+1
A86C  A21C              LDX   #COUNTR-DTAB    ; COUNTR := ABS(GY1-GY2).
A86E  A041              LDY   #GY1-DTAB
A870  20459A            JSR   DMOVI
A873  A047              LDY   #GY2-DTAB
A875  20429C            JSR   DSUBI
A878  20FF9C            JSR   DABSI
A87B  4CA3A8            JMP   :DR060          ; END.

A87E  A21E      :DR050  LDX   #ENDPT-DTAB     ; ELSE BEGIN.
A880  A015              LDY   #DELTAC-DTAB    ; ENDPT := DELTAC.
A882  20459A            JSR   DMOVI
A885  A218              LDX   #ROWAC-DTAB     ; ROWAC := DELTAC/2.
A887  20459A            JSR   DMOVI
A88A  4699              LSR   ROWAC+1
A88C  6698              ROR   ROWAC
A88E  A900              LDA   #0              ; COLAC := 0.
A890  859A              STA   COLAC
A892  859B              STA   COLAC+1
A894  A21C              LDX   #COUNTR-DTAB    ; COUNTR := ABS(GX1-GX2).
A896  A03E              LDY   #GX1-DTAB
A898  20459A            JSR   DMOVI
A89B  A044              LDY   #GX2-DTAB
A89D  20429C            JSR   DSUBI
A8A0  20FF9C            JSR   DABSI           ; END.

A8A3  A59C      :DR060  LDA   COUNTR          ; IF COUNTR>0 THEN BEGIN.
A8A5  059D              ORA   COUNTR+1
A8A7  D003 ^A8AC        BNE   :DR60F
A8A9  4C30AA            JMP   :DR900

A8AC  A218      :DR60F  LDX   #ROWAC-DTAB     ; ROWAC := ROWAC+DELY.
A8AE  A04C              LDY   #DELY-DTAB
A8B0  20329C            JSR   DADDI
A8B3  A01E              LDY   #ENDPT-DTAB     ; IF ROWAC>=ENDPT THEN BEGIN.
A8B5  20229C            JSR   DSCMI
A8B8  9027 ^A8E1        BCC   :DR063

A8BA  20429C            JSR   DSUBI           ; ROWAC := ROWAC-ENDPT.
A8BD  C654              DEC   ROWCRS          ; ROWCRS := ROWCRS-1.
A8BF  A554              LDA   ROWCRS
A8C1  C9FF              CMP   #-1
A8C3  D065 ^A92A        BNE   :DR070          ; ROWCRS >= MINROW.

A8C5  8DD605            STA   HITEDG          ; SET EDGE HIT FLAG.
A8C8  AD5E05            LDA   EDGRUL          ; OFF TOP EDGE.
A8CB  C901              CMP   #EWRAP          ; WRAP?
A8CD  D007 ^A8D6        BNE   :DR061          ; NO -- MUST BE BOUNCE OR HALT.

A8CF  ADAB05            LDA   MAXROW          ; WRAP TO SCREEN BOTTOM EDGE.
A8D2  8554              STA   ROWCRS
A8D4  D054 ^A92A        BNE   :DR070          ; (BRA).

A8D6  E654      :DR061  INC   ROWCRS          ; BRING TURTLE BACK IN.
A8D8  C902              CMP   #EHALT          ; HALT?
A8DA  D030 ^A90C        BNE   :DR067          ; NO.
```

```
A8DC  8DD705              STA    HALTFG          ; YES -- SET FLAG.
A8DF  F049 ^A92A          BEQ    :DR070          ; (BRA).

A8E1  A595        :DR063  LDA    ROWAC+1         ; ELSE IF ROWAC < 0 THEN BEGIN.
A8E3  1045 ^A92A          BPL    :DR070

A8E5  20329C              JSR    DADDI           ; ROWAC := ROWAC+ENDPT.
A8E8  E654                INC    ROWCRS          ; ROWCRS := ROWCRS+1; END.

A8EA  ADAB05              LDA    MAXROW
A8ED  C554                CMP    ROWCRS
A8EF  B039 ^A92A          BCS    :DR070          ; ROWCRS <= MAXROW.

A8F1  8DD605              STA    HITEDG          ; SET EDGE HIT FLAG.
A8F4  AD5E05              LDA    EDGRUL          ; OFF BOTTOM EDGE.
A8F7  C901                CMP    #EWRAP          ; WRAP?
A8F9  D006 ^A901          BNE    :DR065          ; NO -- MUST BE BOUNCE OR HALT.

A8FB  A900                LDA    #0              ; WRAP TO SCREEN TOP EDGE.
A8FD  8554                STA    ROWCRS
A8FF  F029 ^A92A          BEQ    :DR070          ; (BRA).

A901  C654        :DR065  DEC    ROWCRS          ; BRING TURTLE BACK IN.
A903  C902                CMP    #EHALT          ; HALT?
A905  D005 ^A90C          BNE    :DR067          ; NO.

A907  8DD705              STA    HALTFG          ; YES -- SET FLAG.
A90A  F01E ^A92A          BEQ    :DR070          ; (BRA).

A90C  20F19C      :DR067  JSR    DNEGI           ; ROWAC:= ENDPT-ROWAC-1.
A90F  20329C              JSR    DADDI
A912  20129D              JSR    DDCRI

A915  38                  SEC                    ; THETA := THETA-180.
A916  A9B4                LDA    # LOW 180
A918  E5F2                SBC    THETA
A91A  85F2                STA    THETA
A91C  A900                LDA    # HIGH 180
A91E  E5F3                SBC    THETA+1
A920  85F3                STA    THETA+1
A922  2096AB              JSR    MOD360

A925  A24C                LDX    #DELY-DTAB      ; DELY := -DELY.
A927  20F19C              JSR    DNEGI

A92A  A21A        :DR070  LDX    #COLAC-DTAB     ; COLAC := COLAC+DELX.
A92C  A04A                LDY    #DELX-DTAB
A92E  20329C              JSR    DADDI
A931  A01E                LDY    #ENDPT-DTAB     ; IF COLAC >= ENDPT THEN BEGIN.
A933  20229C              JSR    DSCNI
A936  9035 ^A96D          BCC    :DR073

A938  20429C              JSR    DSUBI           ; COLAC := COLAC-ENDPT.
A93B  2043B2              JSR    INCCOL          ; COLCRS := COLCRS+1.
A93E  ADAD05              LDA    MAXCOL+1
A941  C556                CMP    COLCRS+1
```

```
A943  D005 ^A94A          BNE    :DR70F

A945  ADAC05              LDA    MAXCOL
A948  C555                CMP    COLCRS
A94A  B06B ^A9B7  :DR70F  BCS    :DR080          ; COLCRS <= MAXCOL.

A94C  8DD605              STA    HITEDG          ; SET EDGE HIT FLAG.
A94F  AD5E05              LDA    EDGRUL          ; OFF RIGHT EDGE.
A952  C901                CMP    #EWRAP          ; WRAP?
A954  D008 ^A95E          BNE    :DR071          ; NO -- MUST BE BOUNCE OR HALT.

A956  A900                LDA    #0              ; WRAP SCREEN LEFT EDGE.
A958  8555                STA    COLCRS
A95A  8556                STA    COLCRS+1
A95C  F059 ^A9B7          BEQ    :DR080          ; (BRA).

A95E  204AB2      :DR071  JSR    DECCOL          ; BRING TURTLE BACK IN.
A961  AD5E05              LDA    EDGRUL
A964  C902                CMP    #EHALT          ; HALT?
A966  D039 ^A9A1          BNE    :DR077          ; NO.

A968  8DD705              STA    HALTFG          ; YES -- SET FLAG.
A96B  F04A ^A9B7          BEQ    :DR080          ; (BRA).

A96D  A59B        :DR073  LDA    COLAC+1         ; ELSE IF COLAC < 0 THEN BEGIN.
A96F  1046 ^A9B7          BPL    :DR080

A971  20329C              JSR    DADDI           ; COLAC := COLAC+ENDPT.
A974  204AB2              JSR    DECCOL          ; COLCRS := COLCRS-1.

A977  A556                LDA    COLCRS+1
A979  103C ^A9B7          BPL    :DR080          ; COLCRS >= MINCOL.

A97B  8DD605              STA    HITEDG          ; SET EDGE HIT FLAG.
A97E  AD5E05              LDA    EDGRUL          ; OFF LEFT EDGE.
A981  C901                CMP    #EWRAP          ; WRAP?
A983  D00D ^A992          BNE    :DR075          ; NO -- MUST BE BOUNCE.

A985  ADAC05              LDA    MAXCOL          ; WRAP TO SCREEN RIGHT EDGE.
A988  8555                STA    COLCRS
A98A  ADAD05              LDA    MAXCOL+1
A98D  8556                STA    COLCRS+1
A98F  4CB7A9              JMP    :DR080

A992  2043B2      :DR075  JSR    INCCOL          ; BRING TURTLE BACK IN.
A995  AD5E05              LDA    EDGRUL
A998  C902                CMP    #EHALT          ; HALT?
A99A  D005 ^A9A1          BNE    :DR077          ; NO.

A99C  8DD705              STA    HALTFG          ; YES -- SET FLAG.
A99F  F016 ^A9B7          BEQ    :DR080          ; (BRA).

A9A1  20F19C      :DR077  JSR    DNEGI           ; COLAC:=ENDPT-COLAC-1.
A9A4  20329C              JSR    DADDI
A9A7  20129D              JSR    DDCRI

A9AA  A272                LDX    ATHETA-DTAB     ; THETA := -THETA.
```

```
A9AC  20F19C                JSR    DNEGI
A9AF  2096AB                JSR    MOD360
A9B2  A24A                  LDX    #DELX-DTAB        ; DELX := -DELX.
A9B4  20F19C                JSR    DNEGI

A9B7  AD0705       :DR080   LDA    HALTFG            ; HALT?
A9BA  F005 ^A9C1            BEQ    :DR081            ; NO.

A9BC  8D0605                STA    HITEDG            ; YES -- SET EDGE HIT FLAG.
A9BF  D06F ^AA30            BNE    :DR900            ; STOP DRAWING (BRA).

A9C1  ADCD05       :DR081   LDA    WALLS             ; WALLS ACTIVE?
A9C4  0DCE05                ORA    WALLS+1
A9C7  F00D ^A9D6            BEQ    :DR082            ; NO.

A9C9  2067AA                JSR    SGSTUF            ; SAVE GROW & GCOL.
A9CC  207CB2                JSR    TSTPIX            ; GET PIXEL VALUE AT CURRENT POSITION.
A9CF  2028AC                JSR    WALLCK            ; IS IT A WALL?
A9D2  D056 ^AA2A            BNE    :DR300            ; YES -- BACKUP TO PRIOR POSITION.

A9D4  F003 ^A9D9            BEQ    :DR084            ; (BRA)

A9D6  209FAA       :DR082   JSR    CNVRT             ; ROW/COLUMN TO MEM ADDRESS.

A9D9  ADD405       :DR084   LDA    GROPR             ; 'GO'.
A9DC  C905                  CMP    #GO
A9DE  F03F ^AA1F            BEQ    :DR085            ; YES -- DON'T PLOT INTERMEDIATE POINT.

A9E0  2093AA                JSR    PLOT              ; PLOT POINT IF PEN DOWN.

A9E3  ADD405                LDA    GROPR             ; 'FILL' OR 'FILLTO'?
A9E6  2910                  AND    #$10
A9E8  F035 ^AA1F            BEQ    :DR085            ; NO.

A9EA  A900                  LDA    #0                ; YES -- SETUP FOR TSTPIX CALL.
A9EC  8D9C05                STA    FLDCLR
A9EF  A555                  LDA    COLCRS            ; SAVE CURRENT CURSOR POSITION.
A9F1  48                    PHA
A9F2  A556                  LDA    COLCRS+1
A9F4  48                    PHA

A9F5  2093B2       :DR84D   JSR    TSTCOL            ; SEE IF TURTLE AT RIGHT EDGE.
A9F8  AD9F05                LDA    COLFLG
A9FB  2940                  AND    #$40
A9FD  D006 ^AA05            BNE    :DR84E            ; YES

A9FF  2043B2                JSR    INCCOL            ; NO.
AA02  4C0BAA                JMP    :DR84F

AA05  A900         :DR84E   LDA    #0                ; SET TURTLE TO LEFT EDGE.
AA07  8555                  STA    COLCRS
AA09  8556                  STA    COLCRS+1

AA0B  207CB2       :DR84F   JSR    TSTPIX            ; IS TURTLE OVER BACKGROUND?
AA0E  D006 ^AA16            BNE    :DR84M            ; NO -- ALL DONE WITH SCAN.

AA10  2093AA                JSR    PLOT              ; YES -- REPLACE WITH FILL COLOR.
```

```
AA13  4CF5A9                JMP     :DR84D

AA16  68          :DR84M    PLA
AA17  8556                  STA     COLCRS+1
AA19  68                    PLA
AA1A  8555                  STA     COLCRS
AA1C  209FAA                JSR     CNVRT             ; REESTABLISH VISIBLE TURTLE.

AA1F  A21C        :DR085    LDX     #COUNTR-DTAB      ; COUNTR := COUNTR-1.
AA21  20129D                JSR     DDCRI
AA24  2048AA                JSR     SPDDEL
AA27  4C43A8                JMP     :DR060            ; END.

AA2A  8DD505      :DR300    STA     HITWLL            ; SET FLAG.
AA2D  207AAA                JSR     RGSTUF            ; RESTORE GROW & GCOL.

AA30  ADD405      :DR900    LDA     GROPR             ; GO?
AA33  C905                  CMP     #GO
AA35  D010 ^AA47            BNE     :DR990            ; NO.

AA37  ADD505                LDA     HITWLL            ; WALL HIT?
AA3A  D00B ^AA47            BNE     :DR990            ; YES -- DON'T PLOT POINT.

AA3C  ADD605                LDA     NOPLOT            ; PLOT INHIBIT.
AA3F  D006 ^AA47            BNE     :DR990            ; YES -- DON'T PLOT POINT.

AA41  209FAA                JSR     CNVRT             ; PLOT STOP POINT.
AA44  2093AA                JSR     PLOT

AA47  60          :DR990    RTS

AA48                        PROC

AA48  AE5D05      SPDDEL    LDX     SPEED             ; CHECK SPEED SELECTION.
AA4B  F00E ^AA5B            BEQ     :SD200            ; FULL SPEED AHEAD.

AA4D  A514        :SD100    LDA     RTCLOK+2          ; COUNT CLOCK TICKS.

AA4F  C514        :SD110    CMP     RTCLOK+2          ; WAIT FOR ONE TICK.
AA51  F0FC ^AA4F            BEQ     :SD110

AA53  2022AC                JSR     GABRTC            ; OPERATOR BREAK?
AA56  F004 ^AA5C            BEQ     :SD300            ; YES.

AA58  CA                    DEX                       ; DONE?
AA59  D0F2 ^AA4D            BNE     :SD100            ; NO.

AA5B  60          :SD200    RTS

AA5C  20E9A7      :SD300    JSR     SETCR2            ; SET CURSOR.
AA5F  205BA5                JSR     TRTPLC            ; PLACE TURTLE.
AA62  A987                  LDA     #ABTERR
AA64  4C3A7A                JMP     PSTOP

AA67  ADC105      SGSTUF    LDA     GROW              ; YES -- SAVE PRIOR POSITION.
AA6A  8DA005                STA     SAVROW
AA6D  ADCF05                LDA     GCOL
```

```
AA70  8DA105              STA     SAVCOL
AA73  ADD005              LDA     GCOL+1
AA76  8DA205              STA     SAVCOL+1
AA79  60                  RTS

AA7A  ADA005     RGSTUF   LDA     SAVROW          ; RESTORE PRIOR POSITION.
AA7D  8DD105              STA     GROW
AA80  EED905              INC     TUFLAG
AA83  ADA105              LDA     SAVCOL
AA86  8DCF05              STA     GCOL
AA89  ADA205              LDA     SAVCOL+1
AA8C  8DD005              STA     GCOL+1
AA8F  CED905              DEC     TUFLAG
AA92  60                  RTS

AA93  AD1305     PLOT     LDA     PEN             ; PEN UP?
AA96  3006 ^AA9E          BMI     :PL090          ; YES -- DON'T PLOT POINT.

AA98  8D9B05              STA     FCOLOR
AA9B  2056B2              JSR     FPLOT

AA9E  60         :PL090   RTS



AA9F                      PROC
                 ;
                 ; CONVERT ROW/COLUMN CURSOR INTO REAL ADDRESS (FROM SAVMSC ON UP).
                 ;
AA9F  A201       CNVRT    LDX     #01
AAA1  8EAE05              STX     MLTTMP          ; VERTICAL CALCULATIONS.
AAA4  CA                  DEX                     ; VERTICAL CALCULATIONS.
AAA5  86F7                STX     ADRESS+1        ; CLEAR HI BYTE.
AAA7  A554                LDA     ROWCRS          ; ADRESS := ROWCRS*5.
AAA9  8DD105              STA     GROW            ; FOR VISIBLE TURTLE.
AAAC  0A                  ASL     A               ; MULTIPLY BY 4.
AAAD  26F7                ROL     ADRESS+1
AAAF  0A                  ASL     A
AAB0  26F7                ROL     ADRESS+1        ; CLEAR CARRY.
AAB2  6554                ADC     ROWCRS          ; ADD TO MAKE *5.
AAB4  85F6                STA     ADRESS
AAB6  9002 ^AABA          BCC     :CNVR0
AAB8  E6F7                INC     ADRESS+1
AABA  AC3705     :CNVR0   LDY     GSMODE          ; GET MODE
AABD  BEBCB6              LDX     DHLINE,Y        ; GET NUMBER OF SHIFTS.

AAC0  06F6       :CNVR1   ASL     ADRESS          ; ADRESS := ADRESS *X.
AAC2  26F7                ROL     ADRESS+1        ; DO THE DIVIDE.
AAC4  CA                  DEX
AAC5  D0F9 ^AAC0          BNE     :CNVR1

AAC7  A556                LDA     COLCRS+1        ; HORIZONTAL CALCULATIONS.
AAC9  EED905              INC     TUFLAG          ; SET INTERLOCK FOR GCOL UPDATE.
AACC  8DD005              STA     GCOL+1          ; FOR VISIBLE TURTLE.
AACF  4A                  LSR     A               ; SAVE LSB FOR LATER.
AAD0  A555                LDA     COLCRS          ; GET LOW BYTE.
```

```
AAD2  8DCF05                STA     GCOL             ; FOR VISIBLE TURTLE.
AAD5  CECF905               DEC     TUFLAG           ; CLEAR INTERLOCK.
AAD8  BE3083                LDX     DIV2TB,Y         ; GET SHIFT AMOUNT.
AADB  F007 ^AAE4            BEQ     :CNVR3           ; CARRY CLEAR IF NO SHIFT.

AADD  6A           :CNVR2   ROR     A                ; ROLL IN THE CARRY.
AADE  0EAE05                ASL     MLTTMP           ; SHIFT INDEX.
AAE1  CA                    DEX
AAE2  D0F9 ^AADD            BNE     :CNVR2

AAE4  65F6         :CNVR3   ADC     ADRESS           ; CARRY IS ALWAYS CLEAR.
AAE6  9002 ^AAEA            BCC     :CNVR4
AAE8  E6F7                  INC     ADRESS+1
AAEA  18           :CNVR4   CLC
AAEB  6558                  ADC     SAVMSC
AAED  85F6                  STA     ADRESS
AAEF  A5F7                  LDA     ADRESS+1
AAF1  6559                  ADC     SAVMSC+1
AAF3  85F7                  STA     ADRESS+1
AAF5  BE3083                LDX     DIV2TB,Y
AAF8  BDCCB8                LDA     HMASK,X
AAFB  2555                  AND     COLCRS
AAFD  6DAE05                ADC     MLTTMP
AB00  A8                    TAY                      ; MAKE A NEW INDEX.
AB01  B940B3                LDA     DMASKT,Y         ; GET THE FINAL MASK.
AB04  8DB105                STA     DMASK
AB07  8DB005                STA     SHFAMT
AB0A  A000                  LDY     #00              ; SET Y TO ZERO.
AB0C  60                    RTS


AB0D                        PROC
                   ;
                   ; INTEST -- TEST FOR POINT WITHIN SCREEN LIMITS.
                   ;
                   ; CALLING SEQUENCE:
                   ;
                   ;        X = DTAB OFFSET TO X,Y PAIR (EACH TRIPLE PRECISION)
                   ;
                   ;        JSR     INTEST
                   ;        BEQ     POINT IN SCREEN
                   ;
                   ;        A = EDGE TEST BITS (0000LRBT), WHERE 1=OUT OF BOUNDS FOR THAT EDGE.
                   ;
AB0D  84A9         INTEST   STY     TEMP2+2          ; SAVE Y REGISTER.
AB0F  A027                  LDY     #TEMP2-DTAB
AB11  A900                  LDA     #0               ; INITIALIZE RESULT BYTE.
AB13  48                    PHA
AB14  85A8                  STA     TEMP2+1
AB16  B581                  LDA     DTAB+1,X         ; CHECK SIGN OF POSITION.
AB18  3011 ^AB2B            BMI     :IT010           ; NEGATIVE -- COULDN'T BE BEYOND RIGHT EDGE.

AB1A  AD6105                LDA     XC               ; SETUP RIGHT EDGE X POSITION.
AB1D  85A7                  STA     TEMP2
AB1F  20229C                JSR     DBCMI            ; TEST RIGHT EDGE.
AB22  901A ^AB3E            BCC     :IT020           ; INSIDE SCREEN.
AB24  F018 ^AB3E            BEQ     :IT020
```

```
AB26  68                PLA                     ; OUTSIDE -- SET STATUS BIT.
AB27  0904              ORA     #ERIGHT
AB29  D012 ^AB3D        BNE     :IT019          ; (BRA).

AB2B  38         :IT010 SEC                     ; SET UP LEFT EDGE POSITION.
AB2C  A900              LDA     #0
AB2E  ED6105            SBC     XC
AB31  85A7              STA     TEMP2
AB33  C6A8              DEC     TEMP2+1
AB35  20229C            JSR     DSCMI           ; TEST LEFT EDGE.
AB38  B004 ^AB3E        BCS     :IT020          ; INSIDE.

AB3A  68                PLA                     ; OUTSIDE -- SET STATUS BIT.
AB3B  0908              ORA     #ELEFT

AB3D  48         :IT019 PHA

AB3E  E8         :IT020 INX                     ; ADVANCE TO Y POSITION.
AB3F  E8                INX
AB40  E8                INX
AB41  A900              LDA     #0
AB43  85A8              STA     TEMP2+1
AB45  B581              LDA     DTAB+1,X        ; CHECK SIGN OF POSITION.
AB47  1014 ^AB5D        BPL     :IT030          ; POSITIVE -- COULDN'T BE BELOW BOTTOM EDGE.

AB49  38                SEC                     ; SET UP BOTTOM EDGE POSITION.
AB4A  A900              LDA     #0
AB4C  ED6305            SBC     YC
AB4F  85A7              STA     TEMP2
AB51  C6A8              DEC     TEMP2+1
AB53  20229C            JSR     DSCMI           ; TEST BOTTOM EDGE.
AB56  B015 ^AB6D        BCS     :IT040          ; INSIDE.

AB58  68                PLA                     ; OUTSIDE -- SET STATUS BIT.
AB59  0902              ORA     #EBOTOM
AB5B  D00F ^AB6C        BNE     :IT039          ; (BRA).

AB5D  AD6305     :IT030 LDA     YC              ; SETUP TOP EDGE POSITION.
AB60  85A7              STA     TEMP2
AB62  20229C            JSR     DSCMI           ; TEST TOP EDGE.
AB65  9006 ^AB6D        BCC     :IT040          ; INSIDE.
AB67  F004 ^AB6D        BEQ     :IT040

AB69  68                PLA                     ; OUTSIDE -- SET STATUS BIT.
AB6A  0901              ORA     #ETOP

AB6C  48         :IT039 PHA

AB6D  A4A9       :IT040 LDY     TEMP2+2         ; RESTORE Y REGISTER.
AB6F  68                PLA                     ; GET STATUS BYTE FOR EXIT.
AB70  60                RTS



AB71                    PROC
                 ;
```

```
                        ; NEWDEL -- COMPUTE SLOPE DELTAS.
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       JSR     NEWDEL
                        ;       BVS     OVERFLOW
                        ;
                        ;       DELX := GX1-GX2.
                        ;       DELY := GY1-GY2.
                        ;
AB71  A24A              NEWDEL  LDX     #DELX-DTAB      ; DELX := GX1-GX2.
AB73  A03E                      LDY     #GX1-DTAB
AB75  20459A                    JSR     DMOVI
AB78  A044                      LDY     #GX2-DTAB
AB7A  20429C                    JSR     DSUBI
AB7D  700C ^AB8B                BVS     :ND092

AB7F  A24C                      LDX     #DELY-DTAB      ; DELY := GY1-GY2.
AB81  A041                      LDY     #GY1-DTAB
AB83  20459A                    JSR     DMOVI
AB86  A047                      LDY     #GY2-DTAB
AB88  20429C                    JSR     DSUBI

AB8B  60                :ND092  RTS



AB8C                            PROC
                        ;
                        ; NEWCUR -- MOVE NEW CURSOR TO CURRENT CURSOR.
                        ;
                        ;       'GX'    := 'GXNEW'
                        ;       'GY'    := 'GYNEW'
                        ;
AB8C  A206              NEWCUR  LDX     #6              ; 2 VARIABLES OF 3 BYTES EACH.

AB8E  B5E5              :NC010  LDA     GXNEW-1,X
AB90  95E8                      STA     GX-1,X
AB92  CA                        DEX
AB93  D0F9 ^AB8E                BNE     :NC010

AB95  60                        RTS



AB96                            PROC
                        ;
                        ; MOD360 -- 'THETA' = 'THETA' MODULO 360
                        ;
AB96  A5F3              MOD360  LDA     THETA+1         ; SEE IF ANGLE IS NEGATIVE.
AB98  1022 ^ABBC                BPL     :MD020          ; NO.

AB9A  A272                      LDX     #THETA-DTAB     ; YES.
AB9C  20F19C                    JSR     DNEGI           ; GET ABSOLUTE VALUE.
AB9F  A5F3                      LDA     THETA+1         ; THETA = 32768 IS A SPECIAL CASE.
ABA1  3043 ^ABE6                BMI     :MD030
```

```
ABA3  2096AB                JSR    MOD360          ; *** RECURSIVE CALL ***

ABA6  A5F2                  LDA    THETA           ; TEST FOR RESULT = 0.
ABA8  05F3                  ORA    THETA+1
ABAA  F052 ^ABFE            BEQ    :MD099          ; YES -- DONE.

ABAC  A968                  LDA    # LOW 360       ; NO -- THETA = 360 - MOD(ABS(THETA)).
ABAE  38                    SEC
ABAF  E5F2                  SBC    THETA
ABB1  85F2                  STA    THETA
ABB3  A901                  LDA    # HIGH 360
ABB5  E5F3                  SBC    THETA+1
ABB7  85F3                  STA    THETA+1
ABB9  4CEEAB                JMP    :MD090

ABBC  A5F3          :MD020  LDA    THETA+1         ; COMPARE WITH 360.
ABBE  C901                  CMP    # HIGH 360
ABC0  D004 ^ABC6            BNE    :MD025

ABC2  A5F2                  LDA    THETA
ABC4  C968                  CMP    # LOW 360

ABC6  9026 ^ABEE    :MD025  BCC    :MD090          ; THETA < 360.

ABC8  A968                  LDA    # LOW 360       ; PREPARE TO DIVIDE BY 360.
ABCA  85A7                  STA    TEMP2
ABCC  A901                  LDA    # HIGH 360
ABCE  85A8                  STA    TEMP2+1

ABD0  84A9                  STY    TEMP2+2
ABD2  A272                  LDX    #THETA-DTAB
ABD4  A027                  LDY    #TEMP2-DTAB
ABD6  20879C                JSR    DDIVI
ABD9  A4A9                  LDY    TEMP2+2

ABDB  A5A1                  LDA    TEMP            ; REMAINDER IN 'TEMP' AFTER DIVIDE.
ABDD  85F2                  STA    THETA
ABDF  A5A2                  LDA    TEMP+1
ABE1  85F3                  STA    THETA+1
ABE3  4CEEAB                JMP    :MD090

ABE6  A960          :MD030  LDA    # LOW 352       ; -32768 MOD 360 = 352
ABE8  85F2                  STA    THETA
ABEA  A901                  LDA    # HIGH 352
ABEC  85F3                  STA    THETA+1

ABEE  EED905        :MD090  INC    TUFLAG          ; INTERLOCK FOR GANGLE UPDATE.
ABF1  A5F2                  LDA    THETA
ABF3  8DD205                STA    GANGLE
ABF6  A5F3                  LDA    THETA+1
ABF8  8DD305                STA    GANGLE+1
ABFB  CED905                DEC    TUFLAG          ; CLEAR INTERLOCK.

ABFE  60            :MD099  RTS
```

```
ABFF                          PROC
                      ;
                      ; SETCUR -- SET HANDLER CURSOR
                      ;
                      ; CALLING SEQUENCE:
                      ;
                      ;       Y = DTAB OFFSET TO TRIPLE PRECISION X,Y POSITION.
                      ;
                      ;       JSR     SETCUR
                      ;
ABFF  B98200          SETCUR  LDA     DTAB+2,Y
AC02  2A                      ROL     A
AC03  B98000                  LDA     DTAB,Y
AC06  6D6105                  ADC     XC
AC09  8555                    STA     COLCRS
AC0B  B98100                  LDA     DTAB+1,Y
AC0E  6D6205                  ADC     XC+1
AC11  8556                    STA     COLCRS+1

AC13  B98500                  LDA     DTAB+5,Y
AC16  4980                    EOR     #SB
AC18  2A                      ROL     A
AC19  AD6305                  LDA     YC
AC1C  F98300                  SBC     DTAB+3,Y
AC1F  8554                    STA     ROWCRS

AC21  60                      RTS



AC22                          PROC
                      ;
                      ; GABRTC -- GRAPHICS OPERATOR ABORT CHECKER
                      ;
                      ; CALLING SEQUENCE:
                      ;
                      ;       JSR     GABRTC
                      ;       BEQ     ABORT
                      ;
AC22  A511            GABRTC  LDA     BREAK           ; OPERATOR ABORT?
AC24  D004 ^AC2A              BNE     :GA090          ; NO.

AC26  C611                    DEC     BREAK           ; YES -- RESET FLAG.
AC28  A900                    LDA     #0              ; SET EXIT STATUS.

AC2A  60              :GA090  RTS

AC2B                          PROC
                      ;
                      ; WALLCK -- CHECKS TO SEE IF PIXEL VALUE IS A WALL.
                      ;
                      ; CALLING SEQUENCE:
                      ;
                      ;       A =     PIXEL VALUE (00-$0F)
                      ;
                      ;       JSR     WALLCK
                      ;         BNE     PIXEL IS A WALL
                      ;
AC2B  0A              WALLCK    ASL     A                   ; X2.
AC2C  AA                        TAX
AC2D  F00E ^AC3D                BEQ     :WL090              ; BACKGROUND CAN'T BE A WALL.

AC2F  BD3EAC                    LDA     WMASK,X
AC32  2DC005                    AND     WALLS
AC35  D006 ^AC3D                BNE     :WL090              ; FOUND US A WALL.

AC37  BD3FAC                    LDA     WMASK+1,X
AC3A  2DCE05                    AND     WALLS+1

AC3D  60              :WL090    RTS                         ; RETURN WITH CC SET.

AC3E  0000010002      WMASK     DW      0,$01,$02,$04,$08,$10,$20,$40,$80
AC50  0001000200                DW      $100,$200,$400,$800,$1000,$2000,$4000


AC5E                            PROC
                      ;
                      ; GREAD -- READ GRAPHICS DATA FROM SCREEN.
                      ;
                      ; CALLING SEQUENCE:
                      ;
                      ;         CURSOR ALREADY SET TO LOCATION TO READ.
                      ;
                      ;         JSR     GREAD
                      ;
                      ;         A = VALUE OF PIXEL AT CURSOR LOCATION.
                      ;         C = 0 IF TURTLE ON SCREEN, 1 IF OFF.
                      ;
AC5E  A592            GREAD     LDA     EXEC                ; EXECUTE MODE?
AC60  F032 ^AC94                BEQ     :GR090              ; NO.

AC62  AD1405                    LDA     CRFLAG              ; YES -- GRAPHICS SCREEN?
AC65  F02D ^AC94                BEQ     :GR090              ; NO.

AC67  A206                      LDX     #6

AC69  B5EB            :GR010    LDA     GX-1,X              ; ROUND GX TO GX1 ...
AC6B  2A                        ROL     A                   ; ... & GY TO GY1.
AC6C  B5E9                      LDA     GX-3,X
AC6E  6900                      ADC     #0
AC70  95BB                      STA     GX1-3,X
AC72  B5EA                      LDA     GX-2,X
AC74  6900                      ADC     #0
AC76  95BC                      STA     GX1-2,X
AC78  CA                        DEX
AC79  CA                        DEX
AC7A  CA                        DEX
AC7B  D0EC ^AC69                BNE     :GR010

AC7D  A23E                      LDX     #GX1-DTAB           ; YES -- CHECK FOR POINT IN SCREEN LIMITS.
AC7F  200DAB                    JSR     INTEST
AC82  D010 ^AC94                BNE     :GR090              ; NOT IN LIMITS -- RETURN VALUE OF ZERO.
```

```
AC84  8CD805            STY    GRTEMP        ; SAVE Y REGISTER.
AC87  A03E              LDY    #GX1-DTAB     ; SET CURSOR POSITION.
AC89  20FFA8            JSR    SETCUR

AC8C  207C82            JSR    TSTPIX        ; GET PIXEL VALUE.
AC8F  ACD805            LDY    GRTEMP
AC92  18                CLC
AC93  60                RTS

AC94  A900      :GR090  LDA    #0            ; RETURN VALUE OF ZERO.
AC96  38                SEC
AC97  60        VTSRET  RTS

AC98                    PROC
AC98  A900      VTSENS  LDA    #0            ; ASSUME NO OBSTACLE INITIALLY.
AC9A  8D5005            STA    TRTSNS
AC9D  AD1405            LDA    GRFLAG        ; GRAPHICS MODE?
ACA0  F0F5 ^AC97        BEQ    VTSRET        ; NO -- ALL DONE?

ACA2  2067AA            JSR    SGSTUF        ; SAVE GCOL & GROW.
ACA5  98                TYA                  ; SAVE Y REGISTER.
ACA6  48                PHA
ACA7  AD1305            LDA    PEN           ; SAVE PEN.
ACAA  48                PHA
ACAB  AD5E05            LDA    EDGRUL        ; SAVE EDGE RULE.
ACAE  48                PHA
ACAF  A206              LDX    #6            ; SAVE TURTLE LOCATION.

ACB1  B5EB      :ST010  LDA    GX-1,X
ACB3  48                PHA
ACB4  CA                DEX
ACB5  D0FA ^ACB1        BNE    :ST010

ACB7  ADCD05            LDA    WALLS         ; SAVE WALL SELECTIONS.
ACBA  48                PHA
ACBB  ADCE05            LDA    WALLS+1
ACBE  48                PHA

ACBF  AE4E05            LDX    ESTKP         ; ANYTHING IN EXPSTK?
ACC2  F006 ^ACCA        BEQ    :ST017        ; NO.

ACC4  B592      :ST015  LDA    EXPSTK-1,X    ; YES -- SAVE IT ALL.
ACC6  48                PHA
ACC7  CA                DEX
ACC8  D0FA ^ACC4        BNE    :ST015

ACCA  A900      :ST017  LDA    #0            ; CLEAR WALLS.
ACCC  8DCD05            STA    WALLS
ACCF  8DCE05            STA    WALLS+1
ACD2  A980              LDA    #PCUP         ; SET PEN TO UP.
ACD4  8D1305            STA    PEN
ACD7  AD5E05            LDA    EDGRUL        ; IF EDGE RULE = HALT, CHANGE TO FREE.
ACDA  C902              CMP    #EHALT
ACDC  D005 ^ACE3        BNE    :ST020

ACDE  A900              LDA    #EFREE
ACE0  8D5E05            STA    EDGRUL
```

```
ACE3  A905          :ST020  LDA     #GO               ; SIMULATE A GO 1.
ACE5  8D0405                STA     GRDER
ACE8  A900                  LDA     #0
ACEA  8594                  STA     EXPSTK+1
ACEC  A901                  LDA     #1
ACEE  8593                  STA     EXPSTK
ACF0  2032A2                JSR     CALCEL
ACF3  207AA6                JSR     GMOVE

ACF6  AE4E05                LDX     ESTKP             ; RESTORE EXPSTK?
ACF9  F00B ^AD06            BEQ     :ST023            ; NO.

ACFB  A200                  LDX     #0

ACFD  68            :ST022  PLA
ACFE  9593                  STA     EXPSTK,X
AD00  E8                    INX
AD01  EC4E05                CPX     ESTKP
AD04  D0F7 ^ACFD            BNE     :ST022

AD06  68            :ST023  PLA
AD07  8DCE05                STA     WALLS+1
AD0A  68                    PLA
AD0B  8DCD05                STA     WALLS
AD0E  205EAC                JSR     GREAD
AD11  B008 ^AD1B            BCS     :ST025            ; NOT IN SCREEN.

AD13  202BAC                JSR     WALLCK            ; WALL?
AD16  F009 ^AD21            BEQ     :ST030            ; NO

AD18  CE5005                DEC     TRTSNS
AD1B  EE5005        :ST025  INC     TRTSNS            ; YES -- SET SENSOR.
AD1E  EE5005                INC     TRTSNS

AD21  A200          :ST030  LDX     #0                ; RESTORE TURTLE POSITION.

AD23  68            :ST040  PLA
AD24  95EC                  STA     GX,X
AD26  95E6                  STA     GXNEW,X
AD28  E8                    INX
AD29  E006                  CPX     #6
AD2B  D0F6 ^AD23            BNE     :ST040

AD2D  68                    PLA                       ; RESTORE EDGE RULE.
AD2E  8D5E05                STA     EDGRUL
AD31  68                    PLA                       ; RESTORE PEN.
AD32  8D1305                STA     PEN
AD35  207AAA                JSR     RGSTUF            ; RESTORE GCOL & GROW.
AD38  68                    PLA
AD39  A8                    TAY                       ; RESTORE Y-REGISTER.
AD3A  60                    RTS

AD3B                        PROC
                    ;
                    ; SINVAL -- GET VALUE OF SIN(THETA+A*90)
                    ;
                    ; CALLING SEQUENCE:
                    ;
                    ;        A = QUADRANT OFFSET (0-3)
                    ;        'THETA' = ANGLE (0-359)
                    ;
                    ;        JSR      SINVAL
                    ;
                    ;        'TEMP' = SIN(THETA + A*90)
                    ;
AD3B  8543          SINVAL  STA      TEMP+2          ; SAVE QUADRANT OFFSET.
AD3D  84A4                  STY      TEMP+3
AD3F  A072                  LDY      #THETA-DTAB     ; 'ACC' = 'THETA'.
AD41  20A29D                JSR      DLOADA          ; X = 'ACC' - 'DTAB'.

AD44  A95A                  LDA      # LOW 90        ; 'TEMP' = 90.
AD46  85A1                  STA      TEMP
AD48  A900                  LDA      # HIGH 90
AD4A  85A2                  STA      TEMP+1

                    ; NORMALIZE THETA TO 0 - 90 RANGE AND USE TRIG EQUALITIES TO COMPUTE SINE.

AD4C  A021          :SN010  LDY      #TEMP-DTAB      ; IS 'ACC' <= 90.
AD4E  20159C                JSR      DCMPI
AD51  F009 ^AD5C            BEQ      :SN020          ; YES.
AD53  9007 ^AD5C            BCC      :SN020          ; YES.

AD55  E6A3                  INC      TEMP+2          ; NOT YET -- INCREMENT QUADRANT.
AD57  20429C                JSR      DSUBI           ; 'ACC' = 'ACC' - 90.
AD5A  D0F0 ^AD4C            BNE      :SN010          ; (BRA UNLESS RESULT = 0).

AD5C  A6E2          :SN020  LDX      ACC             ; RESULT IS 0 TO 90 FOR TABLE LOOKUP.
AD5E  A5A3                  LDA      TEMP+2          ; QUADRANT #.
AD60  2903                  AND      #$03            ; MODULO 4.
AD62  F018 ^AD7C            BEQ      :SN100          ; QUADRANT 0.

AD64  C901                  CMP      #1
AD66  D008 ^AD70            BNE      :SN040

AD68  A95A                  LDA      #90             ; QUADRANT 1.
AD6A  E5E2                  SBC      ACC
AD6C  AA                    TAX
AD6D  4C7CAD                JMP      :SN100

AD70  C902          :SN040  CMP      #2
AD72  F020 ^AD94            BEQ      :SN150          ; QUADRANT 2.

AD74  A95A                  LDA      #90             ; QUADRANT 3.
AD76  E5E2                  SBC      ACC
AD78  AA                    TAX
AD79  4C94AD                JMP      :SN150

AD7C  A900          :SN100  LDA      #0              ; GET VALUE FROM TABLE.
AD7E  E057                  CPX      #87             ; 87 THRU 90?
AD80  9008 ^AD8A            BCC      :SN120          ; NO -- USE TABLE.

AD82  85A1                  STA      TEMP            ; SPECIAL CASE -- FORCE TO 1.0.
AD84  A901                  LDA      #1
```

```
AD86  85A2                  STA    TEMP+1
AD88  D012 ^AD9C            BNE    :SN900          ; (BRA).

AD8A  85A2         :SN120   STA    TEMP+1          ; MSB = 0.
AD8C  BD9FAD                LDA    SINTAB,X
AD8F  85A1                  STA    TEMP            ; LSB = VALUE FROM TABLE.
AD91  4C9CAD                JMP    :SN900

AD94  207CAD       :SN150   JSR    :SN100          ; GET VALUE TO 'TEMP' *** RECURSIVE CALL ***.
AD97  A221                  LDX    #TEMP-DTAB      ; THEN NEGATE VALUE.
AD99  20F19C                JSR    DNEGI

AD9C  A4A4         :SN900   LDY    TEMP+3
AD9E  60                    RTS



                   ; SINE TABLE VALUES FOR 0 THROUGH 86 DEGREES

         = AD9F    SINTAB = *          ; SIN(X) * 256           X

AD9F  0004090D12            DB    0,4,9,13,18              ; 0-4
ADA4  161B1F2428            DB    22,27,31,36,40           ; 5-9
ADA9  2C31353A3E            DB    44,49,53,58,62           ; 10-14
ADAE  42474B4F53            DB    66,71,75,79,83           ; 15-19
ADB3  585C606468            DB    88,92,96,100,104         ; 20-24
ADB8  6C7074787C            DB    108,112,116,120,124      ; 25-29
ADBD  8084888B8F            DB    128,132,136,139,143      ; 30-34
ADC2  93969A9EA1            DB    147,150,154,158,161      ; 35-39
ADC7  A5A8ABAFB2            DB    165,168,171,175,178      ; 40-44
ADCC  B5B8BBBEC1            DB    181,184,187,190,193      ; 45-49
ADD1  C4C7CACCCF            DB    196,199,202,204,207      ; 50-54
ADD6  D2D4D7D9DB            DB    210,212,215,217,219      ; 55-59
ADDB  DEE0E2E4E6            DB    222,224,226,228,230      ; 60-64
ADE0  E8EAECEDEF            DB    232,234,236,237,239      ; 65-69
ADE5  F1F2F3F5F6            DB    241,242,243,245,246      ; 70-74
ADEA  F7F8F9FAFB            DB    247,248,249,250,251      ; 75-79
ADEF  FCFDFEFEFF            DB    252,253,254,254,255      ; 80-84
ADF4  FFFF                  DB    255,255                  ; 85-86


ADF6                        PROC
                   ;
                   ; TMULT -- TRIPLE PRECISION MULTIPLY
                   ;
                   ; CALLING SEQUENCE:
                   ;
                   ;        'EXPSTK' = WORD OF SIGNED DATA
                   ;        'TEMP' = WORD OF SIGNED DATA
                   ;
                   ;        JSR     TMULT
                   ;
                   ;        'GNUMB'+1 = MSB OF RESULT
                   ;        'GNUMB'+0 = MIDDLE OF RESULT
                   ;        'GNUMB'+2 = LSB OF RESULT
```

```
                          ;
ADF6  A900          TMULT   LDA     #0              ; CLEAR RESULT REGISTER.
ADF8  8D4905                STA     GNUMB
ADFB  8D4A05                STA     GNUMB+1
ADFE  8D4B05                STA     GNUMB+2
AE01  85A6                  STA     TEMP+5          ; SIGN EXTENSION BYTES.
AE03  85A5                  STA     TEMP+4

AE05  A5A2                  LDA     TEMP+1          ; EXTEND SIGN OF 'TEMP'.
AE07  1002 ^AE0B            BPL     :TM005          ; SIGN IS POSITIVE.

AE09  C6A6                  DEC     TEMP+5          ; SIGN IS NEGATIVE.

AE0B  A594          :TM005  LDA     EXPSTK+1        ; EXTEND SIGN OF 'EXPSTK'.
AE0D  1002 ^AE11            BPL     :TM008          ; SIGN IS POSITIVE.

AE0F  C6A5                  DEC     TEMP+4          ; SIGN IS NEGATIVE.

AE11  A218          :TM008  LDX     #24             ; SETUP LOOP COUNT.

AE13  06A1          :TM010  ASL     TEMP
AE15  26A2                  ROL     TEMP+1
AE17  26A6                  ROL     TEMP+5
AE19  9019 ^AE34            BCC     :TM020

AE1B  18                    CLC
AE1C  AD4B05                LDA     GNUMB+2
AE1F  6593                  ADC     EXPSTK
AE21  8D4B05                STA     GNUMB+2
AE24  AD4905                LDA     GNUMB+0
AE27  6594                  ADC     EXPSTK+1
AE29  8D4905                STA     GNUMB+0
AE2C  AD4A05                LDA     GNUMB+1
AE2F  65A5                  ADC     TEMP+4
AE31  8D4A05                STA     GNUMB+1

AE34  CA            :TM020  DEX
AE35  F00C ^AE43            BEQ     :TM090

AE37  0E4B05                ASL     GNUMB+2
AE3A  2E4905                ROL     GNUMB+0
AE3D  2E4A05                ROL     GNUMB+1
AE40  4C13AE                JMP     :TM010

AE43  60            :TM090  RTS


AE44                        PROC
                    ;
                    ; TADDI -- TRIPLE PRECISION ADDITION
                    ;
                    ; CALLING SEQUENCE:
                    ;
                    ;       X = DTAB OFFSET
                    ;
                    ;       JSR     TADDI
```

```
                          ;
                          ;       DTAB(X) = DTAB(X) + 'GNUMB'
                          ;
                          ; NOTE: MSB IS DTAB(X+1), MIDDLE IS DTAB(X+0), LSB IS DTAB(X+2)
                          ;
 AE44  18                 TADDT   CLC
 AE45  B582                       LDA     DTAB+2,X
 AE47  6D4B05                     ADC     GNUMB+2
 AE4A  9582                       STA     DTAB+2,X
 AE4C  B580                       LDA     DTAB+0,X
 AE4E  6D4905                     ADC     GNUMB
 AE51  9580                       STA     DTAB+0,X
 AE53  B581                       LDA     DTAB+1,X
 AE55  6D4A05                     ADC     GNUMB+1
 AE58  9581                       STA     DTAB+1,X
 AE5A  60                         RTS


 AE5B                             PROC
                          ;
                          ; QMULT -- 16 * 16 YIELDING 32 BIT SIGNED MULTIPLY
                          ;
                          ; CALLING SEQUENCE:
                          ;
                          ;       'GACC' = 2 BYTE MULTIPLICAND.
                          ;       Y = DTAB OFFSET TO 2 BYTE MULTIPLIER.
                          ;
                          ;       JSR     QMULT
                          ;
                          ;       'GACC'[4 BYTE] = 'GACC'[2 BYTE] * 'DTAB'(Y)[2 BYTE]
                          ;
 AE5B  A204               QMULT   LDX     #4

 AE5D  B5CD               :QM010  LDA     GACC-1,X
 AE5F  95D1                       STA     GTEMP-1,X
 AE61  A900                       LDA     #0
 AE63  95CD                       STA     GACC-1,X
 AE65  95D5                       STA     GTEMP2-1,X
 AE67  CA                         DEX
 AE68  D0F3 ^AE5D                 BNE     :QM010

 AE6A  B98000                     LDA     DTAB,Y
 AE6D  85D6                       STA     GTEMP2
 AE6F  B98100                     LDA     DTAB+1,Y
 AE72  85D7                       STA     GTEMP2+1
 AE74  1006 ^AE7C                 BPL     :QM015

 AE76  A9FF                       LDA     #-1             ; EXTEND SIGN.
 AE78  85D8                       STA     GTEMP2+2
 AE7A  85D9                       STA     GTEMP2+3

 AE7C  A5D3               :QM015  LDA     GTEMP+1
 AE7E  1004 ^AE84                 BPL     :QM020

 AE80  A9FF                       LDA     #-1             ; EXTEND SIGN.
 AE82  D002 ^AE86                 BNE     :QM022          ; (BRA).
```

```
AE84  A900          :QM020  LDA     #0

AE86  85D4          :QM022  STA     GTEMP+2
AE88  85D5                  STA     GTEMP+3

AE8A  A220                  LDX     #32              ; SETUP LOOP COUNT.

AE8C  06D2          :QM030  ASL     GTEMP            ; LONG SHIFT LEFT.
AE8E  26D3                  ROL     GTEMP+1
AE90  26D4                  ROL     GTEMP+2
AE92  26D5                  ROL     GTEMP+3
AE94  9019 ^AEAF            BCC     :QM040           ; MSB NOT SET.

AE96  18                    CLC                      ; BIT SET -- ADD TO PARTIAL.
AE97  A5CE                  LDA     GACC
AE99  65D6                  ADC     GTEMP2
AE9B  85CE                  STA     GACC
AE9D  A5CF                  LDA     GACC+1
AE9F  65D7                  ADC     GTEMP2+1
AEA1  85CF                  STA     GACC+1
AEA3  A5D0                  LDA     GACC+2
AEA5  65D8                  ADC     GTEMP2+2
AEA7  85D0                  STA     GACC+2
AEA9  A5D1                  LDA     GACC+3
AEAB  65D9                  ADC     GTEMP2+3
AEAD  85D1                  STA     GACC+3

AEAF  CA            :QM040  DEX                      ; DONE?
AEB0  F00B ^AEBD            BEQ     :QM090           ; YES.

AEB2  06CE                  ASL     GACC             ; LONG SHIFT LEFT.
AEB4  26CF                  ROL     GACC+1
AEB6  26D0                  ROL     GACC+2
AEB8  26D1                  ROL     GACC+3
AEBA  4C8CAE                JMP     :QM030

AEBD  60            :QM090  RTS


AEBE                        PROC
                    ;
                    ; QDIV -- 32 DIVIDED BY 16 YIELDING 16 BIT SIGNED DIVIDE.
                    ;
                    ; CALLING SEQUENCE:
                    ;
                    ;       'GACC' = 4 BYTE DIVIDEND.
                    ;       Y = DTAB OFFSET TO 2 BYTE DIVISOR.
                    ;
                    ;       JSR     QDIV
                    ;       BNE     OVERFLOW
                    ;
                    ;       'GACC'[2 BYTE] = 'GACC'[4 BYTE] / 'DTAB'(Y)[2 BYTE]
                    ;       X = DTAB OFFSET TO 'GACC'.
                    ;
      = AEBE        QDIV    = *
```

```
AEBE  B98000                 LDA     DTAB,Y          ; CHECK FOR DIVIDE BY ZERO.
AEC1  198100                 ORA     DTAB+1,Y
AEC4  F066 ^AF2C             BEQ     :QD097

AEC6  A921                   LDA     #32+1           ; LOOP COUNT.
AEC8  85A1                   STA     TEMP

AECA  A900                   LDA     #0
AECC  85D2                   STA     GTEMP           ; CLEAR REMAINDER TO START.
AECE  85D3                   STA     GTEMP+1

AED0  B98100                 LDA     DTAB+1,Y        ; SEE IF DIVISOR IS NEGATIVE.
AED3  85A2                   STA     TEMP+1          ; SAVE FOR LATER.
AED5  1008 ^AEDF             BPL     :QD003          ; NO.

AED7  98                     TYA                     ; YES -- NEGATE DIVISOR ...
AED8  AA                     TAX
AED9  20F19C                 JSR     DNEGI
AEDC  2036AF                 JSR     GNEGA           ; ... & DIVIDEND.

AEDF  A5D1         :QD003    LDA     GACC+3          ; SEE IF DIVIDEND IS NEGATIVE.
AEE1  85A3                   STA     TEMP+2          ; SAVE FOR LATER.
AEE3  1003 ^AEE8             BPL     :QD006          ; NO.

AEE5  2036AF                 JSR     GNEGA           ; YES -- NEGATE IT.

AEE8  A252         :QD006    LDX     #GTEMP-DTAB
AEEA  18                     CLC

AEEB  26CE         :QD010    ROL     GACC            ; LONG ROTATE LEFT.
AEED  26CF                   ROL     GACC+1
AEEF  26D0                   ROL     GACC+2
AEF1  26D1                   ROL     GACC+3
AEF3  26D2                   ROL     GTEMP           ; REMAINDER * 2 + NEW BIT.
AEF5  26D3                   ROL     GTEMP+1

AEF7  C6A1                   DEC     TEMP            ; DONE?
AEF9  F00B ^AF06             BEQ     :QD090          ; YES.

AEFB  20159C                 JSR     DCMPI           ; IS REMAINDER < DIVISOR?
AEFE  90EB ^AEEB             BCC     :QD010          ; YES.

AF00  20429C                 JSR     DSUBI           ; NO -- CORRECT FOR THAT.
AF03  38                     SEC
AF04  B0E5 ^AEEB             BCS     :QD010          ; (BRA).

AF06  20159C       :QD090    JSR     DCMPI
AF09  9007 ^AF12             BCC     :QD091

AF0B  A24E                   LDX     #GACC-DTAB
AF0D  A901                   LDA     #1
AF0F  20049D                 JSR     DADDS

AF12  A5A3         :QD091    LDA     TEMP+2          ; DONE -- SEE IF RESULT IS TO BE NEGATED?
AF14  1003 ^AF19             BPL     :QD093          ; NO.

AF16  2036AF                 JSR     GNEGA           ; YES.
```

```
AF19  A5A2       :GD093  LDA     TEMP+1          ; SEE IF DIVISOR WAS NEGATED AT BEGINNING.
AF1B  1005 ^AF22         BPL     :GD096          ; NO.

AF1D  98                 TYA
AF1E  AA                 TAX
AF1F  20F19C             JSR     DNEGI           ; YES -- CORRECT FOR THAT.

AF22  A24E       :GD096  LDX     #GACC-DTAB      ; AS ADVERTISED.

                 ; CHECK FOR OVERFLOW IN RESULT

AF24  A5CF               LDA     GACC+1          ; CHECK MSB OF USABLE PORTION.
AF26  1009 ^AF31         BPL     :GD098          ; POSITIVE.

AF28  A5D0               LDA     GACC+2
AF2A  25D1               AND     GACC+3

AF2C  C9FF       :GD097  CMP     #-1
AF2E  4C35AF             JMP     :GD099

AF31  A5D0       :GD098  LDA     GACC+2
AF33  05D1               ORA     GACC+3

AF35  60         :GD099  RTS                     ; RETURN WITH CC SET.


AF36                     PROC
                 ;
                 ; QNEGA -- 4 BYTE NEGATE
                 ;
                 ; CALLING SEQUENCE:
                 ;
                 ;       JSR     QNEGA
                 ;
                 ;       'GACC'[4 BYTE] = - 'GACC'[4 BYTE]
                 ;
AF36  38         QNEGA   SEC                     ; CLEAR BORROW.
AF37  A900               LDA     #0
AF39  E5CE               SBC     GACC
AF3B  85CE               STA     GACC

AF3D  A900               LDA     #0
AF3F  E5CF               SBC     GACC+1
AF41  85CF               STA     GACC+1

AF43  A900               LDA     #0
AF45  E5D0               SBC     GACC+2
AF47  85D0               STA     GACC+2

AF49  A900               LDA     #0
AF4B  E5D1               SBC     GACC+3
AF4D  85D1               STA     GACC+3

AF4F  60                 RTS
```

```
AF50                        PROC
                     ;
                     ; RADDI -- DOUBLE PRECISION ADD WITH ROUND FROM FRACTION
                     ;
AF50  B98200         RADDI  LDA     DTAB+2,Y         ; GET FRACTION.
AF53  2A                    ROL     A                ; MSB OF FRACTION TO CARRY.
AF54  4C339C                JMP     DADDIX

AF57                        PROC
                     ;
                     ; RSUBI -- DOUBLE PRECISION SUBTRACT WITH BORROW FROM FRACTION.
                     ;
AF57  B98200         RSUBI  LDA     DTAB+2,Y         ; GET FRACTION.
AF5A  4980                  EOR     #$80             ; INVERT MSB OF FRACTION.
AF5C  2A                    ROL     A                ; INVERTED MSB TO CARRY.
AF5D  4C439C                JMP     DSUBIX


AF60                        PROC
                     ; GPINIT -- INITIALIZE GRAPHICS PARAMETERS (X, Y, THETA & PEN COLOR)

AF60  A900           GPINIT LDA     #0               ; PEN = ERASE AND DOWN.
AF62  8D1305                STA     PEN

AF65  A910                  LDA     #SPLIT           ; FORCE SPLIT SCREEN.
AF67  8D5205                STA     SPLTSC

AF6A  A908                  LDA     #EFREE           ; FREE EDGE TURTLE.
AF6C  8D5E05                STA     EDGRUL

AF6F  A907                  LDA     #SCNMOD          ; SET DEFAULT SCREEN MODE.
AF71  8D3705                STA     GSMODE

AF74  60                    RTS

AF75                        PROC
                     ;
                     ; DFCLRS -- SET DEFAULT COLORS
                     ;
                     ; CALLING SEQUENCE:
                     ;
                     ;      GSMODE     = GRAPHICS MODE
                     ;
                     ;      JSR      DFCLRS
                     ;
                     ;      'PEN'      = 0
                     ;      'NXTCLR'   = 1
                     ;      'PNCLRS'   = DEFAULT VALUES
                     ;      COLOR REGS = DEFAULT VALUES
                     ;
AF75  A200           DFCLRS LDX     #0               ; BACKGROUND ...
AF77  8E1305                STX     PEN
AF7A  A901                  LDA     #CBLACK          ; ... BLACK.
AF7C  20F7A4                JSR     SETCLR
```

```
AF7F  A201               LDX    #1           ; PEN #1 ...
AF81  8E9A05             STX    NXTCLR
AF84  A942               LDA    #CRED        ; ... RED.
AF86  20F7A4             JSR    SETCLR

AF89  A202               LDX    #2           ; PEN #2 ...
AF8B  A91A               LDA    #CYELLO      ; ... YELLOW.
AF8D  20F7A4             JSR    SETCLR

AF90  A203               LDX    #3           ; PEN #3 ...
AF92  A984               LDA    #CBLUE       ; ... BLUE.
AF94  20F7A4             JSR    SETCLR
AF97  60                 RTS
```

```
AF98                  PROC
                 *
                 *  ENTRY POINT FOR FILL ROUTINE:
                 *
                 *  THE FOLLOWING PARAMETERS MUST BE SET ON ENTRY:
                 *
                 *  GSMODE=GRAPHIC MODE INDEX
                 *  FCOLOR=FILL COLOR
                 *  ROWCRS,COLCRS=STARTING COORDINATES
                 *  MAXROW,MAXCOL=MODE DEPENDENT VALUES
                 *  FSTACK = FILL STACK BASE ADDRESS
                 *
AF98             FLOOD                        ;ROUTINE ENTRY POINT
AF98  A580                LDA     S1H                ; INITIALIZE FLOODSTACK POINTER.
AF9A  85F4                STA     FSTACK
AF9C  A581                LDA     S1H+1
AF9E  85F5                STA     FSTACK+1
                 *
                 *  SAVE STARTING COORDINATES
                 *
AFA0  A554                LDA     ROWCRS
AFA2  8DA005              STA     SAVROW
AFA5  A555                LDA     COLCRS
AFA7  8DA105              STA     SAVCOL
AFAA  A556                LDA     COLCRS+1
AFAC  8DA205              STA     SAVCOL+1

AFAF  AE3705              LDX     GSMODE             ; MASK FCOLOR DOWN TO RANGE.
AFB2  AD9805              LDA     FCOLOR
AFB5  3DF6B7              AND     DATMSK,X
AFB8  8D9805              STA     FCOLOR
                 *
                 *  READ DATA AT STARTING COORDINATES
                 *  SAVE AS "FIELD COLOR"
                 *
AFBB  209FAA              JSR     CNVRT              ;GET ADDRESS
                 *
AFBE  ADE105              LDA     DMASK
AFC1  31F6                AND     (ADRESS),Y

AFC3  4EB005     :FIL02   LSR     SHFAMT
AFC6  B003 ^AFCB          BCS     :FIL03
AFC8  4A                  LSR     A
AFC9  90F8 ^AFC3          BCC     :FIL02

AFCB  8D9C05     :FIL03   STA     FLDCLR             ; FIELD COLOR
AFCE  CD9805              CMP     FCOLOR             ; SAME AS FILL COLOR?
AFD1  D003 ^AFD6          BNE     :FIL3D             ; NO.

AFD3  4C26B2              JMP     :FIL90             ; YES -- ALL DONE.

                 *
AFD6  2056B2     :FIL3D   JSR     FPLOT              ;PLOT INITIAL POINT
                 *
                 *
AFD9  2086B2              JSR     TSTROW             ;TEST ROW
```

```
                          *
AFD9  208882                    JSR     TSTROW          ;TEST ROW
```

```
AFDC  2C9E05                    BIT     ROWFLG
AFDF  1006 ^AFE7                BPL     :FIL04          ;NOT ROW 0
                          *
                          *  IF STARTING ROW=0 THEN BEGIN
                          *  ALGORITHM IN THE DOWN DIRECTION
                          *
AFE1  A901                      LDA     #DOWN
AFE3  8597                      STA     ROWINC
AFE5  D004 ^AFEB                BNE     :FIL05
```

```
                         *
                         *  STARTING ROW > 0, BEGIN ALGORITHM
                         *  IN THE UP DIRECTION
                         *
AFE7  A9FF               :FIL04  LDA     #UP
AFE9  8597                       STA     ROWINC
                         *
                         *  PLOT TO STARTING LEFT COLUMN
                         *
AFEB  2093B2             :FIL05  JSR     TSTCOL
AFEE  2C9F05                     BIT     COLFLG
AFF1  3010 ^B003                 BMI     :FIL07          ;COLCRS=0
                         *
AFF3  204AB2                     JSR     DECCOL
AFF6  207CB2                     JSR     TSTPIX
AFF9  D005 ^B000                 BNE     :FIL06
AFFB  2056B2                     JSR     FPLOT
AFFE  B0EB ^AFEB                 BCS     :FIL05
                         *
B000  2043B2             :FIL06  JSR     INCCOL
B003  A555               :FIL07  LDA     COLCRS
B005  8DA305                     STA     LFTCOL
B008  A556                       LDA     COLCRS+1
B00A  8DA405                     STA     LFTCOL+1
                         *
                         * RESET START COLUMN
                         *
B00D  ADA105                     LDA     SAVCOL
B010  8555                       STA     COLCRS
B012  ADA205                     LDA     SAVCOL+1
B015  8556                       STA     COLCRS+1
                         *
                         * FPLOT TO STARTING RIGHT COLUMN
                         *
B017  2093B2             :FIL08  JSR     TSTCOL
B01A  2C9F05                     BIT     COLFLG
B01D  7010 ^B02F                 BVS     :FIL10          ;SCREEN EDGE
                         *
B01F  2043B2                     JSR     INCCOL
B022  207CB2                     JSR     TSTPIX
B025  D005 ^B02C                 BNE     :FIL09
B027  2056B2                     JSR     FPLOT           ;FILL PIXEL
B02A  B0EB ^B017                 BCS     :FIL08
                         *
B02C  204AB2             :FIL09  JSR     DECCOL
B02F  A555               :FIL10  LDA     COLCRS
B031  8DA705                     STA     RGTCOL
B034  A556                       LDA     COLCRS+1
B036  8DA805                     STA     RGTCOL+1
```

```
                   *
                   *  TEST ROW -- IF TOP OR BOTTOM THEN
                   *               NOTHING REQUIRED ON STACK
                   *
B039  20B6B2              JSR     TSTROW
B03C  AD9E05              LDA     ROWFLG
B03F  D00F ^B050          BNE     :FIL11          ;TOP OR BOTTOM
                   *
                   *  PUSH ONTO FILL STACK --
                   *        ROWCRS
                   *        DIRECTION
                   *        LFTCOL
                   *        RGTCOL
                   *
B041  2003B2              JSR     REVROW          ;REVERSE ROW/DIRECTION
                   *
B044  20F1B2              JSR     STKROW
B047  20F5B2              JSR     STKLC
B04A  2001B3              JSR     STKRC
                   *
B04D  2003B2              JSR     REVROW          ;RESTORE
                   *
                   *
                   *  START THE FILL ALGORITHM
                   *
B050  18           :FIL11 CLC                     ;GO TO NEXT ROW
B051  A554                LDA     ROWCRS
B053  6597                ADC     ROWINC
B055  8554                STA     ROWCRS
                   *
B057  2022AC       :FIL12 JSR     GABRTC          ; OPERATOR ABORT?
B05A  D003 ^B05F          BNE     :FIL13          ; NO.
B05C  4C3692              JMP     :FIL95          ; YES.

B05F  AD4305       :FIL13 LDA     LFTCOL
B062  8555                STA     COLCRS
B064  AD4405              LDA     LFTCOL+1
B067  8556                STA     COLCRS+1
                   *
B069  207CB2              JSR     TSTPIX
B06C  D062 ^B0D0          BNE     :FIL20          ;BORDER PIXEL
                   *
B06E  205692       :FIL14 JSR     FPLOT           ;FILL PIXEL
                   *
B071  A555                LDA     COLCRS          ;SAVE NEW
B073  8D4505              STA     NEWLC           ;LEFT
B076  A556                LDA     COLCRS+1        ;COLUMN
B078  8D4605              STA     NEWLC+1
                   *
B07B  2093B2              JSR     TSTCOL
B07E  2C9F05              BIT     COLFLG
B081  3008 ^B08B          BMI     :FIL15          ;LEFT SCREEN EDGE
                   *
B083  20A4B2              JSR     DECCOL
B086  207CB2              JSR     TSTPIX
B089  F0E3 ^B06E          BEQ     :FIL14          ;FIELD PIXEL
```

```
                       *
                       *  BOUNDARY ENCOUNTERED
                       *  COMPARE NEWLC TO LFTCOL
                       *
B08B  38               :FIL15  SEC
B08C  ADA305                   LDA     LFTCOL
B08F  EDA505                   SBC     NEWLC
B092  8595                     STA     DELTAC
B094  ADA405                   LDA     LFTCOL+1
B097  EDA605                   SBC     NEWLC+1
                       *
B09A  D006 ^B0A2               BNE     :FIL16          ;POSSIBLE OPENING
B09C  A595                     LDA     DELTAC
B09E  C903                     CMP     #3
B0A0  9070 ^B112               BCC     :FIL30
                       *
                       *
                       *  POSSIBLE OPENING -- TEST FOR CLOSURE
                       *
B0A2  20D3B2           :FIL16  JSR     REVROW
B0A5  2043B2           :FIL17  JSR     INCCOL
                       *
B0A8  A556                     LDA     COLCRS+1
B0AA  CDA405                   CMP     LFTCOL+1
B0AD  D00D ^B0BC               BNE     :FIL18
B0AF  A555                     LDA     COLCRS
B0B1  CDA305                   CMP     LFTCOL
B0B4  D006 ^B0BC               BNE     :FIL18
                       *
                       *  CLOSURE -- LEFT EDGE FOUND
                       *  NO AREA TO BE PLACED ON STACK
                       *  CONTINUE WITH SEARCH FOR RIGHT EDGE
                       *
B0B6  20D3B2                   JSR     REVROW
B0B9  4C12B1                   JMP     :FIL30
                       *
B0BC  207CB2           :FIL18  JSR     TSTPIX
B0BF  D0E4 ^B0A5               BNE     :FIL17          ;BORDER PIXEL
                       *
                       *  FIELD COLOR FOUND --
                       *  SAVE AREA DEFINITION ON STACK
                       *
B0C1  20E1B2                   JSR     STKROW
B0C4  20E8B2                   JSR     STKCC           ;CURRENT COLCRS
B0C7  20F5B2                   JSR     STKLC           ;LEFT COLUMN
                       *
B0CA  20D3B2                   JSR     REVROW
B0CD  4C12B1                   JMP     :FIL30
```

```
                    *
                    *
                    *  BOUNDARY PIXEL ABOVE/BELOW LEFT COLUMN
                    *  SEARCH RIGHT TO FIND NEW LFTCOL
                    *  IF RGTCOL REACHED W/O FIELD PIXEL,
                    *  THEN AREA IS CLOSED, JUMP TO POP
                    *  STACK
                    *
80D0  ADA405        :FIL20  LDA     LFTCOL+1
80D3  CDA805                CMP     RGTCOL+1
80D6  D00B ^80E3            BNE     :FIL21
80D8  ADA305                LDA     LFTCOL
80DB  CDA705                CMP     RGTCOL
80DE  D003 ^80E3            BNE     :FIL21
                    *
                    *  IF LFTCOL=RGTCOL THEN CLOSURE
                    *
80E0  4CF881                JMP     :FIL70
                    *
80E3  2043B2        :FIL21  JSR     INCCOL
                    *
80E6  207C82                JSR     TSTPIX
80E9  F011 ^80FC            BEQ     :FIL22
                    *
                    *  COMPARE TO RGTCOL
                    *
80EB  A556                  LDA     COLCRS+1
80ED  CDA805                CMP     RGTCOL+1
80F0  D0F1 ^80E3            BNE     :FIL21
80F2  A555                  LDA     COLCRS
80F4  CDA705                CMP     RGTCOL
80F7  D0EA ^80E3            BNE     :FIL21
                    *
80F9  4CF881                JMP     :FIL70           ;CLOSURE
                    *
                    *
                    *  FIELD PIXEL FOUND --

                    *  FILL PIXEL
                    *  SET NEWLC
                    *  PROCEED TO SEARCH RIGHT FOR RGTCOL
                    *
80FC  205682        :FIL22  JSR     FPLOT
80FF  A555                  LDA     COLCRS
8101  8DA505                STA     NEWLC
8104  8DA905                STA     NEWRC
8107  A556                  LDA     COLCRS+1
8109  8DA605                STA     NEWLC+1
810C  8DAA05                STA     NEWRC+1
810F  4C3781                JMP     :FIL34
```

```
                         *
                         *  SEARCH RIGHT FROM LFTCOL TO FIND
                         *  NEW RGTCOL
                         *
B112  ADA305             :FIL30  LDA     LFTCOL
B115  8555                       STA     COLCRS
B117  8DA905                     STA     NEWRC
B11A  ADA405                     LDA     LFTCOL+1
B11D  8556                       STA     COLCRS+1
B11F  8DAA05                     STA     NEWRC+1
                         *
B122  2043B2             :FIL32  JSR     INCCOL
                         *
B125  207CB2                     JSR     TSTPIX
B128  D015 ^B13F                 BNE     :FIL35          ;BORDER PIXEL
                         *
B12A  2056B2                     JSR     FPLOT           ;FILL PIXEL
                         *
B12D  A555                       LDA     COLCRS
B12F  8DA905                     STA     NEWRC
B132  A556                       LDA     COLCRS+1
B134  8DAA05                     STA     NEWRC+1
                         *
B137  2093B2             :FIL34  JSR     TSTCOL
B13A  2C9F05                     BIT     COLFLG
B13D  50E3 ^B122                 BVC     :FIL32          ;NOT RIGHT SCREEN EDGE
                         *
                         *
                         *  NEWRC FOUND -- COMPARE TO RGTCOL
                         *
B13F  ADAA05             :FIL35  LDA     NEWRC+1
B142  CDA805                     CMP     RGTCOL+1
B145  900C ^B153                 BCC     :FIL40
B147  F002 ^B14B                 BEQ     :FIL36
B149  B046 ^B191                 BCS     :FIL50
                         *
B14B  ADA905             :FIL36  LDA     NEWRC
B14E  CDA705                     CMP     RGTCOL
B151  B03E ^B191                 BCS     :FIL50
                         *
                         *  NEWRC < RGTCOL
                         *  IF DELTAC > 3 THEN POSSIBLE OPENING
                         *                     IN SAME DIRECTION
                         *
B153  38                 :FIL40  SEC
B154  ADA705                     LDA     RGTCOL
B157  EDA905                     SBC     NEWRC
B15A  8595                       STA     DELTAC
B15C  ADA805                     LDA     RGTCOL+1
B15F  EDAA05                     SBC     NEWRC+1
B162  D009 ^B16D                 BNE     :FIL41
B164  A595                       LDA     DELTAC
B166  C903                       CMP     #3
B168  B003 ^B16D                 BCS     :FIL41
B16A  4CD5B1                     JMP     :FIL60
```

```
                        *
                        *  CHECK FOR CLOSURE
                        *
B16D  2043B2            :FIL41  JSR     INCCOL
B170  A556                      LDA     COLCRS+1
B172  CDA605                    CMP     RGTCOL+1
B175  D009 ^B180                BNE     :FIL43
B177  A555                      LDA     COLCRS
B179  CDA705                    CMP     RGTCOL
B17C  F002 ^B180                BEQ     :FIL43
B17E  B00E ^B18E                BCS     :FIL49  ;CLOSURE
                        *
B180  207CB2            :FIL43  JSR     TSTPIX
B183  D0E8 ^B16D                BNE     :FIL41
                        *
                        *  OPENING FOUND -- PUSH AREA
                        *    DEFINITION ON THE STACK
                        *
B185  20E1B2                    JSR     STKROW
B188  20EBB2                    JSR     STKCC   ;CURRENT COLCRS
B18B  2001B3                    JSR     STKRC   ;RIGHT COLUMN
                        *
B18E  4CD5B1            :FIL49  JMP     :FIL60
                        *
                        *
                        *  NEWRC >= RGTCOL
                        *  IF DELTAC > 3 THEN POSSIBLE OPENING
                        *                   IN THE OPPOSITE DIRECTION
                        *
B191  F042 ^B1D5        :FIL50  BEQ     :FIL60
B193  38                        SEC
B194  ADA905                    LDA     NEWRC
B197  EDA705                    SBC     RGTCOL
B19A  8595                      STA     DELTAC
B19C  ADAA05                    LDA     NEWRC+1
B19F  EDA605                    SBC     RGTCOL+1
B1A2  D006 ^B1AA                BNE     :FIL51
B1A4  A595                      LDA     DELTAC
B1A6  C903                      CMP     #3
B1A8  9028 ^B1D5                BCC     :FIL60
```

```
                        *
                        *  POSSIBLE OPENING - CHECK FOR CLOSURE
                        *
B1AA  20D3B2            :FIL51  JSR     REVROW
B1AD  204AB2            :FIL52  JSR     DECCOL
                        *
B1B0  A556                      LDA     COLCRS+1
B1B2  CDA805                    CMP     RGTCOL+1
B1B5  D00D ^B1C4                BNE     :FIL53
B1B7  A555                      LDA     COLCRS
B1B9  CDA705                    CMP     RGTCOL
B1BC  D006 ^B1C4                BNE     :FIL53
                        *
                        *  CLOSURE
                        *
B1BE  20D3B2                    JSR     REVROW
B1C1  4CD5B1                    JMP     :FIL60
                        *
B1C4  207CB2            :FIL53  JSR     TSTPIX
B1C7  D0E4 ^B1AD                BNE     :FIL52
                        *
                        *  FIELD COLOR FOUND -- PUSH AREA
                        *   DEFINITION ON THE STACK
                        *
B1C9  20E1B2                    JSR     STKROW
B1CC  2001B3                    JSR     STKRC    ;RIGHT COLUMN
B1CF  20EBB2                    JSR     STKCC    ;CURRENT COLCRS
                        *
B1D2  20D3B2                    JSR     REVROW
                        *
                        *
                        *  CURRENT ROW FILLED --
                        *    TEST FOR SCREEN EDGES, IF NOT
                        *    THEN RESET LFTCOL AND RGTCOL
                        *    AND JUMP TO START OF ALGORITHM
                        *
B1D5  20B8B2            :FIL60  JSR     TSTROW
B1D8  AD9E05                    LDA     ROWFLG
B1DB  D01B ^B1F8                BNE     :FIL70          ;SCREEN TOP OR BOTTOM
                        *
B1DD  ADA505                    LDA     NEWLC
B1E0  8DA305                    STA     LFTCOL
B1E3  ADA605                    LDA     NEWLC+1
B1E6  8DA405                    STA     LFTCOL+1
                        *
B1E9  ADA905                    LDA     NEWRC
B1EC  8DA705                    STA     RGTCOL
B1EF  ADAA05                    LDA     NEWRC+1
B1F2  8DA805                    STA     RGTCOL+1
                        *
B1F5  4C5080                    JMP     :FIL11
```

```
                          *
                          *
                          *  CLOSURE DETERMINED --
                          *  POP FILL STACK FOR OTHER AREAS TO
                          *  BE FILLED
                          *
                          *  SETUP NEW ROW,DIRECTION,LFTCOL,RGTCOL
                          *  JUMP TO START OF ALGORITHM
                          *
B1F8  A274                :FIL70  LDX     #FSTACK-DTAB     ; STACK EMPTY TEST.
B1FA  A036                        LDY     #S1H-DTAB
B1FC  20159C                      JSR     DCMPI
B1FF  F025 ^B226                  BEQ     :FIL90           ; DONE.

B201  2012B3                      JSR     POPFS
B204  8DA805                      STA     RGTCOL+1
B207  2012B3                      JSR     POPFS
B20A  8DA705                      STA     RGTCOL
B20D  2012B3                      JSR     POPFS
B210  8DA405                      STA     LFTCOL+1
B213  2012B3                      JSR     POPFS
B216  8DA305                      STA     LFTCOL
B219  2012B3                      JSR     POPFS
B21C  8597                        STA     ROWINC
B21E  2012B3                      JSR     POPFS
B221  8554                        STA     ROWCRS
                          *
B223  4C57B0                      JMP     :FIL12
                          *
                          *
                          *  FILL FUNCTION COMPLETE
                          *
                          *  RESTORE STARTING CURSOR COORDINATES
                          *  AND RETURN
                          *
B226  ADA005              :FIL90  LDA     SAVROW
B229  8554                        STA     ROWCRS
B22B  ADA105                      LDA     SAVCOL
B22E  8555                        STA     COLCRS
B230  ADA205                      LDA     SAVCOL+1
B233  8556                        STA     COLCRS+1
                          *
B235  60                          RTS
                          *
                          ; FILL FUNCTION ABORT
                          ;
B236  A03E                :FIL95  LDY     #GX1-DTAB
B238  20FFAB                      JSR     SETCUR
B23B  209FAA                      JSR     CNVRT
B23E  A987                        LDA     #ARTERR
B240  4C347A                      JMP     PSTOP
```

```
B243                          PROC

                      *
                      *  SUBROUTINES TO SUPPORT THE FILL ROUTINE
                      *
                      *

B243                          PROC

                      *
                      *  INCREMENT COLCRS
                      *
B243  E655            INCCOL  INC     COLCRS
B245  D002 ^B249              BNE     :ICX
B247  E656                    INC     COLCRS+1
B249  60              :ICX    RTS
                      *
                      *  DECREMENT COLCRS
                      *
B24A  38              DECCOL  SEC
B24B  A555                    LDA     COLCRS
B24D  E901                    SBC     #1
B24F  8555                    STA     COLCRS
B251  B002 ^B255              BCS     :DCX
B253  C656                    DEC     COLCRS+1
B255  60              :DCX    RTS
                      *
                      *  PLOT DATA POINT AT ROWCRS,COLCRS
                      *  ADRESS ALREADY SET BY CONVRT
                      *
B256  ADFF02          FPLOT   LDA     SSFLAG          ; HONOR START/STOP (CTRL-1).
B259  D0FB ^B256              BNE     FPLOT

B25B  AD8105                  LDA     DMASK
B25E  8D8005                  STA     SHFAMT
B261  AD9B05                  LDA     FCOLOR          ; FILL COLOR
B264  4E8005          :FPLT1  LSR     SHFAMT
B267  B003 ^B26C              BCS     :FPLT2
B269  0A                      ASL     A
B26A  90F8 ^B264              BCC     :FPLT1          ;UNCONDITIONAL
B26C  8D9D05          :FPLT2  STA     MSKTMP          ;MASKED DATA
B26F  AD8105                  LDA     DMASK
B272  49FF                    EOR     #$FF
B274  31F6                    AND     (ADRESS),Y
B276  0D9D05                  ORA     MSKTMP
B279  91F6                    STA     (ADRESS),Y
B27B  60                      RTS                     ;CARRY SET

B27C                          PROC

                      *
                      *  TSTPIX --
                      *    CONVERT ROW,COL TO ADDRESS
                      *    UNMASK DATA BIT(S)
                      *    COMPARE WITH FIELD COLOR
                      *    RETURN TO TEST CONDITIONS
                      *
B27C  209FAA          TSTPIX  JSR     CNVRT
B27F  ADB105                  LDA     DMASK
B282  8DB005                  STA     SHFAMT
B285  31F6                    AND     (ADRESS),Y
B287  4EB005          :TSTP1  LSR     SHFAMT          ;RIGHT JUSTIFY
B28A  B003 ^B28F              BCS     :TSTP2          ;DATA PIXEL
B28C  4A                      LSR     A
B28D  90F8 ^B287              BCC     :TSTP1          ;UNCONDITIONAL
B28F  CD9C05          :TSTP2  CMP     FLDCLR          ;COMPARE TO FIELD COLOR
B292  60                      RTS
```

```
B293                     PROC
                   *
                   *  TSTCOL -- TEST CURSOR COLUMN
                   *  SET COLFLG=$80 FOR COLUMN=0
                   *  SET COLFLG=$40 FOR COLUMN=MAX
                   *
B293  A900         TSTCOL  LDA     #0
B295  8D9F05               STA     COLFLG
                   *
B298  A556                 LDA     COLCRS+1
B29A  0555                 ORA     COLCRS
B29C  D006 ^B2A4           BNE     :TSTC1
                   *
                   *  COLUMN=0
                   *
B29E  A980                 LDA     #$80
B2A0  8D9F05               STA     COLFLG
B2A3  60                   RTS
                   *
B2A4  A556         :TSTC1  LDA     COLCRS+1
B2A6  CDAD05               CMP     MAXCOL+1
B2A9  D00C ^B2B7           BNE     :TSTC9
B2AB  A555                 LDA     COLCRS
B2AD  CDAC05               CMP     MAXCOL
B2B0  D005 ^B2B7           BNE     :TSTC9
                   *
                   *  COLUMN=MAXCOL (RIGHT SCREEN EDGE)
                   *
B2B2  A940                 LDA     #$40
B2B4  8D9F05               STA     COLFLG
                   *
B2B7  60           :TSTC9  RTS
```

```
B2B8                            PROC
                     *
                     *  TSTROW -- TEST CURSOR ROW
                     *  SET ROWFLG=$80 FOR ROW=0
                     *  SET ROWFLG=$40 FOR ROW=MAX
                     *
B2B8  A900           TSTROW     LDA     #0
B2BA  8D9E05                    STA     ROWFLG
                     *
B2BD  18                        CLC
B2BE  A554                      LDA     ROWCRS
B2C0  D006 ^B2C8                BNE     :TSTR1
                     *
                     *  ROW=0
                     *
B2C2  A980                      LDA     #$80
B2C4  8D9E05                    STA     ROWFLG
B2C7  60                        RTS
                     *
B2C8  CDAB05         :TSTR1     CMP     MAXROW
B2CB  9005 ^B2D2                BCC     TSTRWX

                     *
                     *  ROW=MAXROW (BOTTOM SCREEN EDGE)
                     *
B2CD  A940                      LDA     #$40
B2CF  8D9E05                    STA     ROWFLG
                     *
B2D2  60             TSTRWX     RTS

B2D3                            PROC
                     *
                     *
                     *  REVROW -- REVERSE ROW INCREMENT
                     *            VALUE (CHANGE SIGN)
                     *            AND ADD TO ROWCRS
                     *
B2D3  18             REVROW     CLC
B2D4  A597                      LDA     ROWINC
B2D6  49FF                      EOR     #$FF
B2D8  6901                      ADC     #1
B2DA  8597                      STA     ROWINC
B2DC  6554                      ADC     ROWCRS
B2DE  8554                      STA     ROWCRS
B2E0  60                        RTS
```

```
B2E1                        PROC
                    *
                    *  STACK SUBROUTINES
                    *
                    *  STKROW - PUSH ROWCRS,ROWINC ONTO FILL STACK
                    *
B2E1  A554          STKROW  LDA     ROWCRS
B2E3  201FB3                JSR     PUSHFS
B2E6  A597                  LDA     ROWINC
B2E8  4C1FB3                JMP     PUSHFS
                    *
                    *  STKCC - PUSH CURRENT COLUMN CURSOR ONTO STACK
                    *
B2EB  A555          STKCC   LDA     COLCRS
B2ED  201FB3                JSR     PUSHFS
B2F0  A556                  LDA     COLCRS+1
B2F2  4C1FB3                JMP     PUSHFS
                    *
                    *  STKLC - PUSH LEFT COLUMN ONTO STACK
                    *
B2F5  ADA305        STKLC   LDA     LFTCOL
B2F8  201FB3                JSR     PUSHFS
B2FB  ADA405                LDA     LFTCOL+1
B2FE  4C1FB3                JMP     PUSHFS
                    *
                    *  STKRC - PUSH RIGHT COLUMN ONTO STACK
                    *
B301  ADA705        STKRC   LDA     RGTCOL
B304  201FB3                JSR     PUSHFS
B307  ADA805                LDA     RGTCOL+1
B30A  4C1FB3                JMP     PUSHFS


B30D  A9A4          STKOVF  LDA     #FSOFER
B30F  4C3A7A                JMP     PSTOP
                    ;
                    ; POPFS -- POP ONE BYTE FROM STACK
                    ;
B312  A5F4          POPFS   LDA     FSTACK          ; FSTACK := FSTACK-1.
B314  D002 ^B318            BNE     :POP10

B316  C6F5                  DEC     FSTACK+1

B318  C6F4          :POP10  DEC     FSTACK
B31A  A000                  LDY     #0
B31C  B1F4                  LDA     (FSTACK),Y
B31E  60                    RTS

                    ;
                    ; PUSHFS -- PUSH ONE BYTE TO STACK.
                    ;
B31F  A4F5          PUSHFS  LDY     FSTACK+1
B321  C4B3                  CPY     SZL+1
B323  B0E8 ^B30D            BCS     STKOVF

B325  A000                  LDY     #0
```

```
B327  91F4                  STA    (FSTACK),Y

B329  E6F4                  INC    FSTACK         ; FSTACK := FSTACK+1.
B32B  D002 ^B32F            BNE    :PSH90

B32D  E6F5                  INC    FSTACK+1

B32F  60           :PSH90   RTS
```

```
                        *
                        *  TABLES
                        *
                        *  DIV2TB = NUMBER OF SHIFTS FOR COLUMN CURSOR
                        *           (INDICATES PIXELS PER BYTE)
                        *  DMASKT = TABLE OF PIXEL MASKS
                        *
                        *
                        *      DINDEX    ANTIC MODE   BYTSML   DIV2TB
                        *
                        *        0           2          40       0
                        *        1           6          20       0
                        *        2           7          20       0
                        *        3           8          10       2
                        *        4           9          10       3
                        *        5           A          20       2
                        *        6           B          20       3
                        *        7           D          40       2
                        *        8           F          40       3
                        *        9         GTIA 1       40       1
                        *        A         GTIA 2       40       1
                        *        B         GTIA 3       40       1
                        *        C           4          40       0
                        *        D           5          40       0
                        *        E           C          20       3
                        *        F           E          40       2
                        *
B330  0000000203        DIV2TB  DB      0,0,0,2,3,2,3,2,3,1,1,1,0,0,3,2
                        *
B340  00FFF00F          DMASKT  DB      $00,$FF,$F0,$0F
B344  C0300C03                  DB      $C0,$30,$0C,$03
B348  80402010                  DB      $80,$40,$20,$10
B34C  08040201                  DB      $08,$04,$02,$01
                        *
```

```
B350                         PROC
                    ;
                    ; ROBOT TURTLE SUBCOMMANDS FOR PILOT GRAPHICS.
                    ;

                    ;
                    ; RBINIT -- INITIALIZE 'ROBOT TURTLE'
                    ;

B350  AD0305        RBINIT  LDA     RBVECT+1         ; ROBOT DRIVER INSTALLED?
B353  F033 ^B388            BEQ     :RI099           ; NO.
B355  A920                  LDA     #RBON            ; INITIALIZE.
B357  8DC705                STA     RBTCMD
B35A  4C25B4                JMP     REXEC            ; EXIT THROUGH DRIVER.



                    ;
                    ; RONOFF -- 'ROBOT ON/OFF' SUBCOMMAND.
                    ;

B35D  AD0305        RONOFF  LDA     RBVECT+1         ; ROBOT DRIVER INSTALLED?
B360  F021 ^B383            BEQ     :RN090           ; NO.

B362  A20A                  LDX     #ONOFFX          ; CHECK 'ON' OR 'OFF'.
B364  20AB7C                JSR     SBCMAT
B367  D01C ^B385            BNE     :RN092           ; NOT FOUND -- ERROR.

B369  8A                    TXA                      ; SET CC FOR 'ON'/'OFF'.
B36A  F00F ^B37B            BEQ     :RF020           ; 'OFF'.

                    ; 'ROBOT ON'.

B36C  8EC505                STX     RBTON            ; FLAG 'ROBOT ON'.
B36F  A920                  LDA     #RBON            ; INTERNAL COMMAND.
B371  4C1884                JMP     RXDRIV           ; RETURN THROUGH DRIVER.

                    ; 'ROBOT OFF'.

                    ; *** EXTERNAL ENTRY FROM 'GEXIT' ***.

B374  AD0305        RBTOFF  LDA     RBVECT+1         ; ROBOT DRIVER INSTALLED?
B377  F00F ^B388            BEQ     :RN099           ; NO -- NOP.

B379  A200                  LDX     #0               ; FLAG 'ROBOT OFF'.
B37B  8EC505        :RF020  STX     RBTON
B37E  A900                  LDA     #RBOFF           ; INTERNAL COMMAND.
B380  4C1884                JMP     RXDRIV           ; RETURN THROUGH DRIVER.

B383                :RH090
B383                :RP090
B383                :RE090
B383  A902          :RN090  LDA     #IVCERR          ; NO 'ROBOT' OR 'OFF'.
B385                :RH092
B385                :RP092
B385                :RE092
B385  4C3A7A        :RN092  JMP     RSTOP
```

```
B388               :RI099
B388  60           :RN099  RTS


                   ;
                   ; REYES -- ROBOT 'EYES' SUBCOMMAND.
                   ;
B389  ADC505       REYES   LDA     RBTON           ; IS ROBOT ON?
B38C  F0F5 ^B383           BEQ     :RE090          ; NO -- ERROR.

B38E  A20A                 LDX     #ONOFFX         ; CHECK FOR 'ON' OR 'OFF'.
B390  20487C               JSR     SBCMAT
B393  D0F0 ^B385           BNE     :RE092          ; NOT FOUND -- ERROR.

B395  A901                 LDA     #RBEYES         ; INTERNAL COMMAND.
B397  4C1BB4               JMP     RXDRIV          ; RETURN THROUGH DRIVER.


                   ;
                   ; RPEN -- ROBOT 'RPEN' SUBCOMMAND.
                   ;
B39A  ADC505       RPEN    LDA     RBTON           ; IS ROBOT ON?
B39D  F0E4 ^B383           BEQ     :RP090          ; NO -- ERROR.

B39F  A208                 LDY     #UPDWNX         ; CHECK FOR 'UP' OR 'DOWN'.
B3A1  20487C               JSR     SBCMAT
B3A4  D0DF ^B385           BNE     :RP092          ; NOT FOUND -- ERROR.

B3A6  8A                   TXA                     ; CONVERT TO 0 (UP)/1 (DOWN).
B3A7  A200                 LDX     #0              ; ASSUME UP.
B3A9  C940                 CMP     #PCON
B3AB  D001 ^B3AE           BNE     :RP010          ; UP.
B3AD  E8                   INX                     ; DOWN.

B3AE  A902         :RP010  LDA     #RBPEN          ; INTERNAL COMMAND.
B3B0  4C1BB4               JMP     RXDRIV          ; RETURN THROUGH DRIVER.


                   ;
                   ; RHORN -- ROBOT 'HORN' SUBCOMMAND.
                   ;
                   ; 'RHORN OFF' = 'RHORN 0'.
                   ; 'RHORN ON'  = 'RHORN 1'.

B3B3  ADC505       RHORN   LDA     RBTON           ; IS ROBOT ON?
B3B6  F0CB ^B383           BEQ     :RH090          ; NO -- ERROR.

B3B8  A20A                 LDX     #ONOFFX         ; CHECK FOR 'ON' OR 'OFF'.
B3BA  20487C               JSR     SBCMAT
B3BD  F013 ^B3D2           BEQ     :RH020          ; FOUND IT.
```

```
                          ; NOT 'ON' OR 'OFF' - CHECK FOR 0,1,2.

B3BF  206E81                      JSR     ATOM              ; GET 'HORN' PARAMETER.
B3C2  D0C1 ^B385                  BNE     :RH092            ; ERROR -- RETURN.

B3C4  C902                        CMP     #NUM              ; CHECK FOR NUMBER.
B3C6  D0BB ^B383                  BNE     :RH090            ; NO -- ERROR.

B3C8  A6B9                        LDX     NUMBER+1          ; 0,1,2 VALID.
B3CA  D0B7 ^B383                  BNE     :RH090            ; INVALID.
B3CC  A6B8                        LDX     NUMBER
B3CE  E003                        CPX     #3
B3D0  B0B1 ^B383                  BCS     :RH090            ; INVALID.

B3D2  A903              :RH020    LDA     #RBHORN           ; INTERNAL COMMAND.
B3D4  4C1BB4                      JMP     RXDRIV            ; RETURN THROUGH DRIVER.



                          ;
                          ; RGO -- ROBOT 'GO' SUBCOMMAND.
                          ;
                          ; CALLING SEQUENCE:
                          ;
                          ;         ROBOT TURTLE ON
                          ;         EXPSTK+0,+1 = SIGNED MAGNITUDE.
                          ;         EXECUTE MODE
                          ;
                          ;         JSR     RGO
                          ;

B3D7  A215              RGO       LDX     #EXPSTK+2-DTAB    ; EXPSTK+2,+3 = ABSOLUTE VALUE.
B3D9  A013                        LDY     #EXPSTK-DTAB
B3DB  20459A                      JSR     DMOVI

B3DE  A980                        LDA     #RBFWD            ; ASSUME FORWARD.
B3E0  2496                        BIT     EXPSTK+3          ; NOW CHECK SIGN.
B3E2  1005 ^B3E9                  BPL     RGO010            ; FORWARD IT IS.
B3E4  20F19C                      JSR     DNEGI             ; ABSOLUTE VALUE.
B3E7  A981                        LDA     #RBBACK           ; BACK.

                          ; *** EXTERNAL ENTRY FROM 'RTURN' ***.

B3E9  8DC705            RGO010    STA     RBTCMD            ; INTERNAL COMMAND.
B3EC  A595                        LDA     EXPSTK+2          ; VALUE.
B3EE  8DC805                      STA     RBTPRM
B3F1  A596                        LDA     EXPSTK+3
B3F3  8DC905                      STA     RBTPRM+1
B3F6  4C25B4                      JMP     REXEC             ; RETURN THROUGH DRIVER.



                          ;
                          ; RTURN -- ROBOT 'TURN' SUBCOMMAND.
                          ;
                          ; CALLING SEQUENCE:
                          ;
```

```
                        ;       ROBOT TURTLE ON
                        ;       EXPSTK+0,+1 = SIGNED MAGNITUDE
                        ;       EXECUTE MODE
                        ;
                        ;       JSR     RTURN
                        ;

B3F9  A215              RTURN   LDX     #EXPSTK+2-DTAB  ; EXPSTK+2,+3 = ABSOLUTE VALUE.
B3FB  A013                      LDY     #EXPSTK-DTAB
B3FD  20459A                    JSR     DMOVI

B400  A941                      LDA     #RBRGHT         ; ASSUME RIGHT.
B402  2496                      BIT     EXPSTK+3        ; NOW CHECK SIGN.
B404  10E3 ^B3E9                BPL     RG0010          ; RIGHT IT IS.
B406  20F19C                    JSR     DNEGI           ; ABSOLUTE VALUE.
B409  A940                      LDA     #RBLEFT         ; LEFT.
B40B  D0DC ^B3E9                BNE     RG0010



                        ;
                        ; RRDSNS -- ROBOT 'READ SENSORS'.
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       JSR     RRDSNS
                        ;
                        ;       RBTSNS = SENSOR VALUES.
                        ;       ;      = SNESOR VALUES.
                        ;

B40D  A980              RRDSNS  LDA     #RBFWD          ; 'GO 0' IS A 'NOP'.
B40F  A200                      LDX     #0
B411  8EC905                    STX     RBTPRM+1        ; MSB = 0.
B414  201684                    JSR     RXDRIV          ; UPDATE SENSORS.
B417  ADC605                    LDA     RBTSNS          ; AS ADVERTISED.
B41A  60                        RTS
```

```
B41B                          PROC

                         ;
                         ; RXDRIV -- INTERFACE TO ROBOT DRIVER.
                         ;
                         ; CALLING SEQUENCE:
                         ;
                         ;        A = INTERNAL COMMAND.
                         ;        X = LSB OF INTERNAL PARAMETER.
                         ;
                         ;        JSR RXDRIV
                         ;        RETURN WITH 'BEQ' ONLY IF OPERATION COMPLETED.
                         ;        JUMP TO 'PSTOP' IF 'BREAK' OR LOGIC ERROR.
                         ;
                         ;        Y IS PRESERVED.
                         ;
                         ;        CHECKS 'EXEC' FLAG AND RETURNS 'OK' IF 'FALSE'.
                         ;
B41B  8DC705             RXDRIV  STA     RBTCMD           ; INTERNAL COMMAND.
B41E  A592                       LDA     EXEC             ; EXECUTE?
B420  F01D ^B43F                 BEQ     :RX099           ; NO.
B422  8EC805                     STX     RBTPRM           ; LSB (INTERNAL PARAMETER).
                         ; *S*    JMP     REXEC            ; INTERFACE TO DRIVER.



                         ;
                         ; REXEC -- CALL ROBOT DRIVER.
                         ;
                         ; CALLING SEQUENCE:
                         ;
                         ;        'RBTCMD' = INTERNAL COMMAND BYTE.
                         ;        'RBTPRM' = INTERNAL PARAMETER WORD.
                         ;
                         ;        JSR REXEC
                         ;        Y IS PRESERVED.
                         ;
                         ;        RETURN IF OPERATION COMPLETED.
                         ;        JUMP TO 'PSTOP' IF 'BREAK' OR LOGIC ERROR.
                         ;
B425  84A1               REXEC   STY     TEMP             ; SAVE Y.
B427  2040B4                     JSR     :RX100           ; 'JSR' TO DRIVER.
B42A  204FB4                     JSR     TONES

                         ; Y = 1 (OK); = 128 (BREAK); = 132 (LOGIC ERROR).
                         ; A = ROBOT SENSOR STATE.

B42D  8DC605                     STA     RBTSNS
B430  C080                       CPY     #128
B432  9007 ^B438                 BCC     :RX090           ; OK.

      = 0000                     IF DEBUG
   -                             BEQ     :RX020           ; BREAK.
   -                     :RX010  LDA     #INTERR          ; 'BUG'.
```

```
    -                          BNE     :RX022
                               ENDIF

8434  A987          :RX020     LDA     #ABTERR          ; BREAK.
8436  A4A1          :RX022     LDY     TEMP             ; RESTORE Y.
8438  4C3A7A                   JMP     PSTCP

843B  A4A1          :RX090     LDY     TEMP             ; RESTORE Y.
843D  A900                     LDA     #0               ; SET CC FOR EXIT.
843F                :RX099
843F  60                       RTS

8440                :RX100
        = 0000                 IF DEBUG
    -                          LDA     RBVECT+1         ; ROBOT DRIVER INSTALLED?
    -                          BEQ     :RX010           ; NO -- BUG.
                               ENDIF
8440  ADC705                   LDA     RBTCMD           ; STACK-3.
8443  48                       PHA
8444  ADC805                   LDA     RBTPRM           ; STACK-2.
8447  48                       PHA
8448  ADC905                   LDA     RBTPRM+1         ; STACK-1.
844B  48                       PHA
844C  6C0205                   JMP     (RBVECT)
```

```
B44F                            PROC
                      ;
                      ; AUDIO TONE GENERATION PROCESSOR
                      ;
B44F  A208            TONES     LDX     #AUREGS*2       ; SETUP TO SCAN REGISTER ASSIGN TABLE.

B451  BD1305          :T0010    LDA     AUDIOR-2,X      ; POINTER TO VARIABLE.
B454  85B6                      STA     POINT
B456  1D1405                    ORA     AUDIOR-1,X      ; NULL ENTRY IF RESULT IS ZERO.
B459  F003 ^B45E                BEQ     :T0020

B45B  BD1405                    LDA     AUDIOR-1,X      ; FINISH MOVING NON-NULL POINTER.
      = 0000                    IF      FALSE
   -                            STA     POINT+1
   -                            BMI     :T0020          ; NUMERIC CONSTANT.

   -                            LDY     #0              ; NOW GET VALUE.
   -                            LDA     (POINT),Y
                                ENDIF

B45E  291F            :T0020    AND     #$1F            ; MODULO 32.
B460  A8                        TAY
B461  B971B4                    LDA     AUDTAB,Y        ; GET FREQ FROM TABLE.
B464  9DFED1                    STA     AUDF1-2,X       ; PUT IN HARDWARE.
B467  A9A4                      LDA     #$A4            ; QUARTER VOLUME.
B469  9DFFD1                    STA     AUDC1-2,X
B46C  CA                        DEX
B46D  CA                        DEX
B46E  D0F1 ^B451                BNE     :T0010

B470  60              :T0090    RTS


B471  00              AUDTAB    DB    0                 ; REST
B472  F3E6D9CCC1                DB    243,230,217,204,193,182
B478  ACA2999088                DB    172,162,153,144,136,128
B47E  79726C6660                DB    121,114,108,102,96,91
B484  55514C4844                DB    85,81,76,72,68,64
B48A  3C3935322F                DB    60,57,53,50,47,45
B490  2A                        DB    42
```

```
B491                           PROC
                        ;
                        ; PILVBL -- DEFERRED VBLANK ROUTINE WHICH READS THE
                        ;           CONSOLE KEYS (START/OPTION/SELECT), DEBOUNCES
                        ;           THEM AND RETURNS THE STATUS IN 'CONKEY'.
                        ;
B491  209AB4            PILVBL  JSR     CONKRD          ; READ CONSOLE KEYS.
B494  207AA5                    JSR     TRTLOC          ; VISIBLE TURTLE.
B497  4C62E4                    JMP     XITVBV          ; EXIT VBLANK.

B49A  AD6505            CONKRD  LDA     CSTATE          ; IDLE STATE?
B49D  D01C ^B4BB                BNE     :CK010          ; NO.

B49F  AD1FD0                    LDA     CONSOL          ; YES -- KEY PRESSED?
B4A2  2907                      AND     #ANYKEY
B4A4  C907                      CMP     #ANYKEY
B4A6  F03E ^B4E6                BEQ     :CK090          ; NO -- ALL DONE.

B4A8  4907                      EOR     #ANYKEY         ; INVERT BIT SENSE.
B4AA  8D4405                    STA     CONKEY          ; SAVE FOR 'MLOOP'.
B4AD  EE6505                    INC     CSTATE          ; GO TO STATE 1.
B4B0  A90C                      LDA     #$0C            ; PUT RETURN CODE IN 'CH'.
B4B2  8DFC02                    STA     CH
B4B5  A905                      LDA     #5              ; ACTIVATE TIMER.
B4B7  8D2002                    STA     CDTMV5
B4BA  60                        RTS

B4BB  C901              :CK010  CMP     #1              ; KEY DOWN DEBOUNCE STATE?
B4BD  D009 ^B4C8                BNE     :CK020          ; NO.

B4BF  AD2002                    LDA     CDTMV5          ; YES -- DEBOUNCE DONE?
B4C2  D022 ^B4E6                BNE     :CK090          ; NO.

B4C4  EE6505                    INC     CSTATE          ; GO TO STATE 2.
B4C7  60                        RTS

B4C8  C902              :CK020  CMP     #2              ; WAIT FOR KEYS UP STATE?
B4CA  D012 ^B4DE                BNE     :CK030          ; NO.

B4CC  AD1FD0                    LDA     CONSOL          ; YES -- ALL KEYS UP?
B4CF  2907                      AND     #ANYKEY
B4D1  C907                      CMP     #ANYKEY
B4D3  D011 ^B4E6                BNE     :CK090          ; NO.

B4D5  EE6505                    INC     CSTATE          ; YES -- GO TO STATE 3.

B4D8  A905                      LDA     #5              ; ACTIVATE TIMER.
B4DA  8D2002                    STA     CDTMV5
B4DD  60                        RTS

B4DE  AD2002            :CK030  LDA     CDTMV5          ; DEBOUNCE DONE?
B4E1  D003 ^B4E6                BNE     :CK090          ; NO.

B4E3  8D6505                    STA     CSTATE          ; YES -- GO TO STATE 0.

B4E6  60                :CK090  RTS
```

```
B4E7  48          GRDLI   PHA

B4E8  8D0AD4              STA     WSYNC

B4EB  A984                LDA     #CBLUE
B4ED  454F                EOR     COLRSH          ; ATTRACT
B4EF  254E                AND     DRKMSK
B4F1  8D18D0              STA     COLPF2
B4F4  A91A                LDA     #CYELLO
B4F6  454F                EOR     COLRSH          ; ATTRACT
B4F8  254E                AND     DRKMSK
B4FA  8D17D0              STA     COLPF1

B4FD  68                  PLA
B4FE  40                  RTI
```

```
B4FF                           PROC
                         ;
                         ; MESSOT -- MESSAGE GENERATOR
                         ;
                         ; CALLING SEQUENCE:
                         ;
                         ;         A = MESSAGE # (INDEX TO INTERNAL TABLE)
                         ;
                         ;         JSR     MESSOT
                         ;
        = B4FF           MESSOT = *
B4FF  297F                     AND     #$7F            ; MASK OFF SIGN BIT.
B501  0A                       ASL     A
      = 0000                   IF      DEBUG
  -                            BEQ     :M0100          ; 0 IS ILLEGAL.

  -                            CMP     #MTSIZ+1
  -                            BCS     :M0100          ; # IS TOO LARGE.
                               ENDIF

B502  AA                       TAX
B503  BD34B5                   LDA     MESTAB-2,X      ; GET MESSAGE ADDRESS FROM TABLE.
B506  85A7                     STA     TEMP2
B508  BD35B5                   LDA     MESTAB-1,X
B50B  85A6                     STA     TEMP2+1

B50D  A000                     LDY     #0

B50F  B1A7           :M0010    LDA     (TEMP2),Y
B511  F018 ^B52B               BEQ     :M0090          ; DONE (NO EOL AT END).

B513  C8                       INY                     ; BUMP POINTER.

B514  C90D                     CMP     #CR             ; INTERNAL CR?
B516  D006 ^B51E               BNE     :M0015          ; NO.

B518  20989F                   JSR     NEWLIN
B51B  4C0FB5                   JMP     :M0010          ; CONTINUE.

B51E  C99B           :M0015    CMP     #EOL
B520  F006 ^B528               BEQ     :M0020          ; DONE.

B522  20E294                   JSR     CHOT
B525  4C0FB5                   JMP     :M0010

B528  20989F         :M0020    JSR     NEWLIN

B52B  60             :M0090    RTS

      = 0000                   IF      DEBUG
  -                  :M0100    LSR     A
  -                            STA     ACC             ; PRINT # INSTEAD OF CANNED MESSAGE.
  -                            LDA     #0
  -                            STA     ACC+1
  -                            LDA     #14
  -                            JSR     MESSOT          ; *** RECURSIVE CALL ***
  -                        LDX     #ACC-DTAB
  -                        JSR     DECASC
  -                        JMP     NEWLIN
                           ENDIF

B52C                       PROC
                 ; RDYMES -- GENERATE "READY" MESSAGE SEQUENCE.

B52C  84A9       RDYMES    STY     TEMP2+2          ; SAVE Y REG.
B52E  A901                 LDA     #RDYTXT          ; "READY" TEXT.
B530  20FFB4               JSR     MESSOT
B533  A4A9                 LDY     TEMP2+2
B535  60                   RTS
```

```
B536                          PROC
                     ;
                     ; TABLE OF MESSAGE ADDRESSES
                     ;
B536  90B5           MESTAB   DW    :MES1
B538  97B5                    DW    :MES2
B53A  A4B5                    DW    :MES3
B53C  B8B5                    DW    :MES4
B53E  C7B5                    DW    :MES5
B540  CCB5                    DW    :MES6
B542  D7B5                    DW    :MES7
B544  DDB5                    DW    :MES8
B546  E3B5                    DW    :MES9
B548  E6B5                    DW    :MES10
B54A  F2B5                    DW    :MES11
B54C  97B5                    DW    :MES12
B54E  FEB5                    DW    :MES13
B550  97B5                    DW    :MES14
B552  97B5                    DW    :MES15
B554  12B6                    DW    :MES16
B556  1BB6                    DW    :MES17
B558  25B6                    DW    :MES18
B55A  32B6                    DW    :MES19
B55C  3EB6                    DW    :MES20
B55E  45B6                    DW    :MES21
B560  54B6                    DW    :MES22
B562  62B6                    DW    :MES23
B564  88B6                    DW    :MES24
B566  8EB6                    DW    :MES25
B568  9CB6                    DW    :MES26
B56A  A1B6                    DW    :MES27
B56C  B1B6                    DW    :MES28
B56E  C5B6                    DW    :MES29
B570  CAB6                    DW    :MES30
B572  E0B6                    DW    :MES31
B574  FAB6                    DW    :MES32
B576  1CB7                    DW    :MES33
B578  35B7                    DW    :MES34
B57A  49B7                    DW    :MES35
B57C  5CB7                    DW    :MES36
B57E  75B7                    DW    :MES37
B580  94B7                    DW    :MES38
B582  A7B7                    DW    :MES39
B584  AEB7                    DW    :MES40
B586  BBB7                    DW    :MES41
B588  C6B7                    DW    :MES42
B58A  CFB7                    DW    :MES43
B58C  D6B7                    DW    :MES44
B58E  DEB7                    DW    :MES45
      = 005A         MTSIZ    = *-MESTAB

B590  9B52454144     :MES1     DB    CR,'READY',EOL

      = FFFF                  IF     DEBUG-1
B597                 :MES12
B597                 :MES14
```

```
8597               :MES15
                            ENDIF
8597  5748415427   :MES2    DB   'WHAT',SQUOTE,'S THAT?',0
85A4  43414E2754   :MES3    DB   'CAN',SQUOTE,'T USE COMMAND HERE',0
85BB  4449564944   :MES4    DB   'DIVIDE BY 0',0
85C7  4F4F505300   :MES5    DB   'OOPS',0
85CC  492F4F2045   :MES6    DB   'I/O ERROR ',0
85D7  425245414B   :MES7    DB   'BREAK',0
85DD  202A2A2A20   :MES8    DB   ' *** ',0
85E3  4E4F20524F   :MES9    DB   'NO ROOM',0
85EB  5748455245   :MES10   DB   'WHERE?',0
85F2  553A20544F   :MES11   DB   'U: TOO DEEP',0
       = 0000               IF     DEBUG
 -                 :MES12   DB   'BUG!',0
                            ENDIF
85FE  4C494E4520   :MES13   DB   'LINE # OUT OF RANGE',0
       = 0000               IF     DEBUG
 -                 :MES14   DB   'ERROR #',0
 -                 :MES15   DB   'PLEASE SHORTEN',EOL
                            ENDIF
8612  2420564152   :MES16   DB   '$ VARS:',CR,EOL
861B  0D23205641   :MES17   DB   CR,'# VARS:',CR,EOL
8625  0D55534520   :MES18   DB   CR,'USE STACK:',CR,EOL
8632  0D47522050   :MES19   DB   CR,'GR PARMS:',CR,EOL
863E  5448455441   :MES20   DB   'THETA=',0
8645  0D0D465245   :MES21   DB   CR,CR,'FREE MEMORY=',0
8654  544F4F204D   :MES22   DB   'TOO MANY I/OS',0
8662  7D41544152   :MES23   DB   CLEAR,'ATARI PILOT (C) COPYRIGHT ATARI 1982',EOL
8688  20203E2000   :MES24   DB   '--> ',0
868D  43414E2754   :MES25   DB   'CAN',SQUOTE,'T CONTINUE',0
869C  53544F5000   :MES26   DB   'STOP',0
86A1  0D43414E27   :MES27   DB   CR,'CAN',SQUOTE,'T RENUMBER',EOL
```

```
B6B1  4F5645524C  :MES28   DB   'OVERLAPPING RANGE: ',0
B6C5  20544F2000  :MES29   DB   ' TO ',0
B6CA  0D50524F47  :MES30   DB   CR,'PROGRAM IS UNCHANGED',EOL
B6E0  0D594F5520  :MES31   DB   CR,'YOU ARE ABOUT TO DELETE ',0
B6FA  204C494E45  :MES32   DB   ' LINE(S).',CR,'ARE YOU SURE (Y OR N): ',0
B71C  53504C4954  :MES33   DB   'SPLIT SCREEN NOT ALLOWED',0
B735  4E4F542041  :MES34   DB   'NOT A GRAPHICS MODE',0
B749  0D492F4F20  :MES35   DB   CR,'I/O ASSIGNMENTS:',CR,EOL
B75C  5348414445  :MES36   DB   'SHADE REGION TOO COMPLEX',0
B775  4E4F204D4F  :MES37   DB   'NO MORE PEN COLORS--USE CHANGE',0
B794  414C524541  :MES38   DB   'ALREADY HAVE COLOR',0
B7A7  50454E533A  :MES39   DB   'PENS: ',0
B7AE  4241434B47  :MES40   DB   'BACKGROUND: ',0
B7BB  545552544C  :MES41   DB   'TURTLE PEN: ',0
B7C8  4D4F44453A  :MES42   DB   'MODE: ',0
B7CF  454447453A  :MES43   DB   'EDGE: ',0
B7D6  5350454544  :MES44   DB   'SPEED: ',0
B7DE  57414C4C53  :MES45   DB   'WALLS: ',0
                           EPROC
```

```
                         ;
                         ; GRAPHICS TABLES
                         ;

                         ;
                         ; MODE CHARACTERISTICS (BY MODE)
                         ;
        = 0000           NG      =     0                    ; NOT ALLOWED
        = 0080           FO      =     $80                  ; ALLOWED BUT NO SPLIT SCREEN (FULL ONLY).
        = 0090           SC      =     FO+SPLIT             ; ALLOWED WITH SPLIT SCREEN.

 B7E6   000000           GCHAR   DB    NG,NG,NG
 B7E9   9090909090               DB    SC,SC,SC,SC,SC,SC,FO,FO,FO
 B7F2   00009090                 DB    NG,NG,SC,SC

                         ;
                         ; PIXEL WIDTH MASKS
                         ;

 B7F6   FFFFFF           DATMSK  DB    $FF,$FF,$FF
 B7F9   0301030103               DB    3,1,3,1,3,1,$F,$F,$F
 B802   FFFF0103                 DB    $FF,$FF,1,3

                         ; NUMBER OF FOREGROUND COLORS

 B806   000404           COLRS   DB    0,4,4
 B809   0301030103               DB    3,1,3,1,3,1,15,8,15
 B812   00000103                 DB    0,0,1,3

                         ; SCREEN CENTER OFFSETS

 B816   1300090009       XCENTR  DW    19,9,9
 B81C   1300270027               DW    19,39,39,79,79,159,39,39,39
 B82E   130013004F               DW    19,19,79,79

 B836   06000B0005       YCENTR  DW    11,11,5
 B83C   0B00170017               DW    11,23,23,47,47,95,95,95,95
 B84E   0B0005005F               DW    11,5,95,95

                         ; SCREEN BOUNDARIES FOR FILL

 B856   2600120012       COLMAX  DW    38,18,18
 B85C   26004E004E               DW    38,78,78,158,158,318,78,78,78
 B86E   260026009E               DW    38,38,158,158

 B876   160016000A       ROWMAX  DW    22,22,10
 B87C   16002E002E               DW    22,46,46,94,94,190,190,190,190
 B88E   16000A00BE               DW    22,10,190,190

                         ; TEXT SCREEN MARGINS

 B896   020000           LMRGTB  DB    2,0,0                ; LEFT MARGINS.

 B899   271313           RMRGTB  DB    39,19,19             ; RIGHT MARGINS.

                         ; COLOR CLOCKS PER HORIZONTAL UNIT FOR MODES 0 - 15.
```

```
B89C  080808      CCPXTB  DB   8,8,8
B89F  0402020101          DB   4,2,2,1,1,0,2,2,2  ; (0 = 1/2)
B8A8  04040101            DB   4,4,1,1

                  ; SCAN LINES PER CURSOR VERTICAL UNIT FOR MODES 0 - 15.

B8AC  100810      SLPYTB  DB   16,8,16
B8AF  0804040202          DB   8,4,4,2,2,1,1,1,1  ; (0 = 1/2)
B8B8  08100101            DB   8,16,1,1

                  ; THIS IS THE NUMBER OF LEFT SHIFTS NEEDED TO MULTIPLY COLCRS
                  ; BY # BYTES/ROW.  (ROWCRS*5)/(2*DHLINE)

B8BC  030202      DHLINE  DB   3,2,2
B8BF  0101020203          DB   1,1,2,2,3,3,3,3,3
B8C8  03030203            DB   3,3,2,3

                  ;

B8CC  00010307    HMASK   DB   0,1,3,7

                  ; OFFSETS TO DISPLAY LIST INTERRUPT BYTE FOR SPLIT SCREEN.

B8D0  000000      DLIOFF  DB   0,0,0
B8D3  182C2C5454          DB   24,44,44,84,84,166,0,0,0
B8DC  0000A4A6            DB   0,0,164,166

                  ; VISIBLE TURTLE Y OFFSET

B8E0  0101000100  TRDY    DB   1,1,0,1,0,0
B8E6  0002010101          DB   0,2,1,1,1,0
B8EC  0000020101          DB   0,0,2,1,1,2
B8F2  0000000100          DB   0,0,0,1,0,1

                  ; VISIBLE TURTLE X OFFSET

B8F8  0303030303  TRDX    DB   3,3,3,3,3,3
B8FE  0303030303          DB   3,3,3,3,3,3
B904  0303030304          DB   3,3,3,3,4,4
B90A  0504040404          DB   5,4,4,4,4,4

                  ; VISIBLE TURTLE PLAYER DATA

B910  10383810BA  VTURT   DB   $10,$38,$38,$10,$BA,$FE,$6C,$EE,$FE,$FE,$7C,$7C,$8A,$82     ; 0

      = 000E      VTHITE  = *-VTURT                                                      ; HEIGHT OF TURTLE REP.

B91E  06066E367C          DB   $06,$06,$6E,$36,$7C,$7C,$EE,$EE,$BE,$7F,$7D,$7D,$04,$0C     ; 1

B92C  3313BFFE7E          DB   $33,$13,$BF,$FE,$7E,$EE,$EE,$FF,$FD,$7D,$3C,$18,$08,$18     ; 2

B93A  30133BBFFC          DB   $30,$13,$3B,$BF,$FC,$7E,$EE,$EE,$FF,$7D,$7D,$38,$08,$18     ; 3

B948  18086BFF7D          DB   $18,$08,$6B,$FF,$7D,$EE,$EE,$FE,$7C,$7D,$3B,$20,$60,$00     ; 4

B956  3090FB7FFF          DB   $30,$90,$FB,$7F,$FF,$EE,$EC,$7E,$7C,$32,$46,$C0,$00,$00     ; 5
```

```
 B964  0888807878          DB   $08,$88,$80,$78,$78,$EA,$EF,$FA,$78,$78,$80,$88,$08,$00     ; 6
 B972  0000C04632          DB   $00,$00,$C0,$46,$32,$7C,$7E,$EC,$EE,$FF,$7F,$FB,$90,$30     ; 7
 B980  006020387D          DB   $00,$60,$20,$38,$7D,$7C,$EE,$EE,$FE,$7D,$FF,$BB,$08,$18     ; 8
 B98E  1808387D7D          DB   $18,$08,$38,$7D,$7D,$FF,$EE,$EE,$7E,$FC,$BF,$38,$13,$30     ; 9
 B99C  1808183C7D          DB   $18,$08,$18,$3C,$7D,$FD,$EF,$EE,$FE,$7E,$FE,$EF,$13,$33     ; 10
 B9AA  0C047D7D7F          DB   $0C,$04,$7D,$7D,$7F,$AE,$EE,$FE,$7C,$7C,$36,$6E,$06,$06     ; 11
 B9B8  82BA7C7CFE          DB   $82,$BA,$7C,$7C,$FE,$EE,$EE,$7C,$FE,$BA,$10,$38,$38,$10     ; 12
 B9C6  3020BEBE7E          DB   $30,$20,$BE,$BE,$7E,$6D,$6F,$7F,$3E,$3E,$6C,$76,$60,$60     ; 13
 B9D4  1810183CBE          DB   $18,$10,$18,$3C,$BE,$BF,$FF,$6F,$6F,$7E,$7F,$FD,$C8,$CC     ; 14
 B9E2  18101CBEBE          DB   $18,$10,$1C,$BE,$BE,$FF,$6F,$6F,$7E,$3F,$FD,$DC,$C8,$0C     ; 15
 B9F0  000604DCBE          DB   $00,$06,$04,$DC,$BE,$3E,$77,$77,$7F,$BE,$FF,$DD,$10,$18     ; 16
 B9FE  000003624C          DB   $00,$00,$03,$62,$4C,$3E,$7E,$37,$77,$FF,$FE,$DF,$09,$0C     ; 17
 BA0C  10110D1E1E          DB   $10,$11,$0D,$1E,$1E,$5B,$FB,$5F,$1E,$1E,$0D,$11,$10,$00     ; 18
 BA1A  0C09DFFEFF          DB   $0C,$09,$DF,$FE,$FF,$77,$37,$7E,$3E,$4C,$62,$03,$00,$00     ; 19
 BA28  1810DDFFBE          DB   $18,$10,$DD,$FF,$BE,$77,$77,$7F,$3E,$BE,$DC,$04,$06,$00     ; 20
 BA36  0CC8DCFD3F          DB   $0C,$C8,$DC,$FD,$3F,$7E,$77,$77,$FF,$BE,$BE,$1C,$10,$18     ; 21
 BA44  CCC8FD7F7E          DB   $CC,$C8,$FD,$7F,$7E,$77,$77,$FF,$BF,$BE,$3C,$18,$10,$18     ; 22
 BA52  6060766C3E          DB   $60,$60,$76,$6C,$3E,$3E,$77,$77,$7D,$7E,$BE,$BE,$20,$30     ; 23

                    ; COLOR REGISTER ASSIGNMENTS

 BA60                          PROC
 BA60  80BA           COLADR   DW    :CSET0              ; MODE 0
 BA62  84BA                    DW    :CSET1              ; MODE 1
 BA64  84BA                    DW    :CSET1              ; MODE 2
 BA66  84BA                    DW    :CSET1              ; MODE 3
 BA68  84BA                    DW    :CSET1              ; MODE 4
 BA6A  84BA                    DW    :CSET1              ; MODE 5
 BA6C  84BA                    DW    :CSET1              ; MODE 6
 BA6E  84BA                    DW    :CSET1              ; MODE 7
 BA70  89BA                    DW    :CSET2              ; MODE 8
 BA72  8DBA                    DW    :CSET3              ; MODE 9
 BA74  91BA                    DW    :CSET4              ; MODE 10
 BA76  8DBA                    DW    :CSET3              ; MODE 11
 BA78  80BA                    DW    :CSET0              ; MODE 12
 BA7A  80BA                    DW    :CSET0              ; MODE 13
 BA7C  84BA                    DW    :CSET1              ; MODE 14
 BA7E  84BA                    DW    :CSET1              ; MODE 15
```

```
BA80  00000000     :CSET0  DB    0,0,0,0

BA84  0804050607   :CSET1  DB    8,4,5,6,7             ; BAK, PF0, PF1, PF2 (,PF3 FOR MODES 1 & 2).

BA89  06050000     :CSET2  DB    6,5,0,0               ; PF2, PF1

BA8D  08000000     :CSET3  DB    8,0,0,0               ; BAK

BA91  0001020304   :CSET4  DB    0,1,2,3,4,5,6,7,8

                           EPROC
      = BA99       PRGEND  =     *-1

BA9A                       END      PILINI


no ERRORs, 1789 Labels, $0E00 free.



   ABRTCK 9F7E      24/33   94/49  106/24  107/43  107/47  194#35
   ABTERR 0087       3#29   21/30  194/41  234/56  262/54  275/ 8
   ACC    00E2       7#54  183/23  183/40  183/57  184/19  184/37  244/35  244/44  244/52
   ACCBUF BD7B      10#46   14/10   14/12
   ACCLNG 00FE       4#34   67/58   68/17
   ACLN   0086       7#16   14/11   14/13   66/61   67/ 7   67/12   67/46   67/61   68/30   68/47   68/50   70/59   71/20
                    71/32   71/44   71/45   71/46   71/49   71/56   71/57   72/22   72/28   73/37   73/51   73/53  194/21
                   194/22  194/24
   ACOLR1 0087       7#15   13/27  116/55
   ACOLR2 0086       7#14   13/25  116/53
   ADRESS 00F6       8# 6  102/35  102/37  102/41  102/46  102/47  103/37  103/39  103/44  103/48  103/49  235/38  235/42
                   235/44  235/46  235/48  235/52  235/53  236/15  236/17  236/20  236/21  236/23  253/44  263/49  263/51
                   264/ 9
   AINC   00FC       8# 9   18/46  116/41  120/34  121/ 5  122/22
   AKFLAG 0547       9#21   66/10   66/42   69/41   69/48
   ALINE  00FA       8# 8   17/35   17/37   17/57   18/45  116/30  116/37  120/30  121/ 7  121/25  122/14  122/18  122/21
                   124/33
   ANYKEY 0007       5#48  277/20  277/21  277/24  277/46  277/47
   APPMHI 000E       2#35  136/36  136/39  137/15  137/16
   ASCDEC 9DBB      52/22   69/18  165#20
   ASTMES 0008       3#41   21/19   21/41
   ATMERR 0002       3#26   49/ 9   56/15   92/30  190/47
   ATMRET 83A9      49/17   51/57   54/36   55/10   56#32
   ATOM   816E      17/48   48#24   55/61   66/22   90/10   93/39   95/25   97/55   99/15  125/46  144/22  197/29  199/ 5
                   201/57  272/ 7
   ATOM2  8171      48#28   50/43
   ATRIG  0020       5#59   82/13  146/33  149/ 5
   ATRLIM 0000       5#60  189/16
   ATRNUM 0040       5#58   49/46   51/56
   ATRSTR 0080       5#57   51/43   73/22   81/43   91/30
   ATRTYP 0566       9#39   49/47   51/44   73/23   91/31  146/34  153/36  156/ 6  156/12  156/17  189/17
   AUDC1  D201       4#48   97/26  196/18  276/31
   AUDCLK 9FB4      15/33   97/45  190/45  196# 8
   AUDCTL D208       4#51  196/13
   AUDF1  D200       4#47    4/48   97/25  196/17  276/29
   AUDIUR 0515      8#60   96/41   96/56   97/ 8   97/24  196/19  196/20  276/12  276/14  276/17
   AUDTAB B471    276/28  276#39
   AUREGS 0004      4#38    8/60    8/60   96/ 9  196/15  276/10
   AUTNMS 9169    116/16  116/18  117#15
   AUTOIN 0536      9#12   13/57   16/11   18/25   18/49   18/54  116/52
   AUTOXT 0081      3#19   18/30
   AUX1   051D      8#61   13/58   15/18  139/33  139/44
   AUX2   051E      9# 5   13/59   15/19  139/38  139/45
   AXFLAG 0546      9#20   66/11   67/49   69/39   99/58
n BAKCLR 05BB     10#13
n BELL   00FD      2#14
  BHIGH  0095      8#35  114/57  119/14  119/23  119/26  121/47  123/36  124/ 8  127/38  128/59
  BLOW   0093      8#34  114/56  114/61  115/ 7  119/15  119/24  119/27  121/43  122/ 5  122/53  123/ 6  123/ 9  123/15
                 123/25  124/ 7  127/11  128/54
  BNUM   0097      8#36  118/42  118/43  118/51  120/53  120/54  120/59  122/27  122/30  122/31  127/ 5  127/ 6  127/29
  BOTSCR 02BF      2#49   61/41  134/48
  BPTR   0080      4#11   51/ 5   66/26   68/56   69/24   90/16   90/53   93/42   95/28   97/58   99/19  199/ 8
  BRACKT 93C8    114/50  118/36  120/46  126#56
  BREAK  0011      2#36  194/36  194/39  240/45  240/48
  BSLASH 005C      2#15   37/61   58/26
  BUFPNT 978C    134/ 8  139/50  142#24
  CALDEL A232    207/12  207#24  243/12
  CASCIL 8F9A    107/16  107#26
  CASSGN A4DE    208/22  210/42  217#17
  CASSUF 003C      4#54   15/31  107/26
  CASSON 0034      4#53  107/27
  CBLACK 0001      5#16   40/10  251/60
  CBLUE  0084      5#14   18/56   40/ 8  252/16  279/ 9
  CCPXTB B69C    220/60  286# 6
  CDEST  0530      9# 8   13/ 7   20/27  109/23  109/31  131/23  134/52  135/17  137/34  201/28  201/29  201/31  202/52
                 202/53
  CDG    8078     27/43   31/16   35/60   43# 8
  CDDWN  001D      2# 9  195/56
  CDTAB  803E     27/43   29/18   29/20   31/16   34/38   34/40   34/42   34/44   34/50   34/52   34/54   34/56   34/58
                  34/60   35/ 5   35/ 7   35/ 9   35/11   35/14   35/16   35/18   35/20   35/22   35/24   35/26   35/28
                  35/30   35/32   35/34   35/36   35/38   35/40   35/42   35/44   35/46   35/48   35/50   35/52   35/54
                  35/56   35/58   35/60   36/ 5   36/ 7   36/ 9   36/11   36/13   36/15   36/17   36/19   36/21   36/23
                  36/25   36/27   36/29   36/31   36/33   36/35   42#23   42/34   43/35
  CDTMV5 0220      3#10  277/30  277/36  277/53  277/56
  CETEMP 0583     10# 5  136/27  136/31  136/48  136/50
  CH     02FC      2#37   54/13  277/28
  CHKDEV 9887    139/36  146/56  149#16
  CHKDWS 9F00     66/15   66/35   90/29   91/10  191#54
  CHKLN  946B     17/60  121/10  125/57  129#49
  CHKSLF 9EED    138/54  191# 6
  CHKTKN 9EF9     27/31   29/36   48/28   64/10   80/30   81/10   82/ 7   96/12  100/15  191/12  191#31  201/34  202/14
  CHOT   9482     21/17   59/25   61/31   82/47   82/54   83/30   84/ 5   84/37   84/40   85/ 6   86/49   87/10   87/35
                  87/39   88/26   88/31  104/38  105/31  131#20  142/54  186/49  187/28  187/40  194/ 5  195/29  195/38
                 195/41  195/55  195/57  201/50  202/21  219/10  280/48
 CID    E456      3#19   25/36  132/34  134/ 9  139/52  140/43  141/28
 CKDGRC 9693    148#39  149/18  149/25
 CHECK  9E86     49/20   51/28   78/49   92/12  189#45  190/11  190/44
 CLEAR  0076      2#13   81/30  212/26  263/53
* CLEFT  001E      2#10
 CLETIC 9E91     45/61   70/47  [illegible]  189/45  214/42
 CLMTAC A4BD    214/35  214#30
 CLNCNT 0096     26/32   42#36
```

```
  CLOSE  000C     1#52  140/40
  CLOSEM 9881    62/27   75/17  148# 9
  CLRMAT A496   208/ 9  209/17  209/42  209/56  210/29  211/26  213/58  214/45  215#58
  CLRFRG 87C0    13/40   75/34   75#37  110/18
  CLRTXT A56B   220/13  220#17  220/54
  CMACUM 7CBD    30/47   31/26   32/ 6   32#44
  CMATCH 7C56    27/37   30#39
  CNDERR 0002     3#20   47/38
  CNTERR 0099     3#45   65/11
  CNUMBF 9E83    16/38   26/16   48/49   69/14  185/44  188#21  189/49
  CNVRT  AA9F   210/54  227/58  229/28  233/27  234/11  234/31  235#35  253/41  262/53  264/ 6
  COLAC  009A     8#28    8/29  229/57  229/59  229/60  230/23  230/24  231/51  232/29
  COLADR BA60   217/53  217/55  287#46
  COLCHS 0055     2#45   58/44   61/25  231/61  232/ 8  232/17  232/18  232/35  232/44  232/46  233/41  233/43  233/55
                233/56  234/ 8  234/10  235/57  235/61  236/26  240/20  240/23  253/28  253/30  255/25  255/27  255/33
                255/35  255/50  255/52  256/40  256/42  256/49  256/51  257/27  257/30  258/31  258/34  258/48  258/51
                259/10  259/13  259/23  259/25  260/ 9  260/12  261/11  261/14  262/43  262/45  263/18  263/20  263/26
                263/28  263/30  265/15  265/16  265/25  265/28  267/19  267/21
n COLDST 0244     2#28
  COLFLG 059F     9#49  233/47  255/15  255/40  256/55  259/29  265/13  265/22  265/35
  COLMAX B856   134/28  134/30  285#47
  COLORO 02C4     2#56   18/57   18/59  116/54  116/56
  COLPF1 D017     5# 6  279/16
  COLPF2 D018     5# 7  279/12
  COLPS  B806   134/36  285#31
  COLRSH 004F     2#31  279/10  279/14
  COMBUF BD00    10#45   14/21   15/50   15/52
  COMPRS 9629   112/60  133/ 8  133/38  136#13
  COND   810E    28/52   46#25
  CONKEY 0544     9#18   16/16   16/20   16/22   16/51   16/59   16/61  277/25
  CONKRD B49A   277/12  277#16
  CONSOL D01F     4#56  277/19  277/45
  COUNTR 009C     8#29    8/30  230/ 7  230/25  230/32  230/33  234/13
  CR     000D     2# 6  280/39  282/57  283/39  283/41  283/41  283/43  283/43  283/45  283/45  283/49  283/49  283/61
                284/10  284/12  284/14  284/20  284/20
  CRED   0042     5#13   39/59  252/ 8
  CRIGHT 001F     2#11   71/14
  CRSINH 02F0     2#38  134/14  137/47  147/15  195/53
  CRSNUP 9FA7    99/35   99/55  132/19  132/46  195#53
  CSTATE 0565     9#38  277/16  277/26  277/39  277/50  277/59
  CTAB   7D01    31/22   31/24   34#35  105/ 9  105/10
  CTABAT 0510     8#54   13/61   30/39   30/54   31/30   31/35   47/34
  CTBOTH 0060    34#32   34/33   35/14   35/16   35/22   35/24   35/26   36/31
  CTCLN  0010    34#31   34/33   34/33   47/33
  CTIMR  0040    16/43   34#29   34/32   34/40   34/42   34/44   34/50   34/52   34/54   34/56   34/58   34/60   35/ 5
                 35/ 7   35/ 9   35/11   35/18   35/20
  CTNORM 0060    34#33   34/38   35/28   35/30   35/32   35/34   35/36   35/38   35/40   35/44   35/46   35/48   35/50
                 35/52   35/54   35/56   35/58   35/60   36/ 5   36/ 7   36/ 9   36/11   36/13   36/15   36/17   36/19
                 36/21   36/23   36/27   36/29   36/33   36/35
  CTRUN  0020    18/38   34#30   34/32   35/42   36/25
  CUP    001C     2# 8  195/54
  CYELLO 001A     5#15   18/58   39/61  252/12  279/13
  DABBI  9CFF   177# 6  229/42  229/47  230/12  230/30
  DADDR  9DAC   164/49  163#57
  DADDI  9C32    18/47   44/13  121/ 8  122/23  123/27  123/60  161/45  165/13  166/48  167/25  167/33  173# 7  183/58
                187/16  207/50  230/39  231/12  231/36  231/53  232/32  232/58
  DADDIX 9C33   173# 9  251/13
```

```
DADLP  9D08     33/16  151/40  153/ 7  153/12  153/17  177#32
DADDS  9D04     50/14   83/18  127/30  128/27  177#26  223/14  223/20  223/40  249/56
DAND1  9D68     44/24  180#34
DATMSK B7F6    253/35  285#25
DCAERR 0026      3#57  210/21
DCLOSF 973F     19/14   25/52  101/16  102/52  103/54  104/42  109/28  133/10  133/45  135/58  137/28  139/29  140/21
               140#38  148/11
DCM010 9C1C    171#58  172/28
DCMFA  9DB6    164/52  184#37
DCMFI  9C15     24/44  114/58  121/39  121/44  121/49  122/44  122/54  124/ 9  126/58  127/25  128/55  128/60  158/38
               158/42  164/57  166/38  171#52  175/39  178/11  178/15  184/38  229/51  244/27  249/44  249/51  262/16
DCNCI  9C0F    129/51  171#29  204/23
DDCFI  9D12     17/41   94/44  106/28  120/61  122/28  177#43  231/37  232/59  234/14
DDIVI  9C87     44/15  174#55  176/19  239/38
DEBUG  0000      1#11  199/36  200/ 8  274/59  275/18  280/19  280/55  282/59  283/27  283/33
DECASC 9E14     21/39   82/35   82/50   82/57   83/ 6   83/19   84/33   87/19   88/39   88/55  118/52  124/34  124/38
               186#39  193/44  202/40  218/47
DECCUL B24A    232/21  232/33  255/18  255/49  256/58  261/ 9  263#25
DELMES 009F      3#51    3/52  118/48
DELTAC 0095      8#25    8/26  229/44  229/50  230/16  257/12  257/17  259/52  259/56  260/38  260/42
DELTAR 0093      8#24    8/25  229/39  229/49  229/55
DELX   00CA      8#19    8/20  226/20  227/15  229/45  231/52  233/ 7  238/15
DELY   00CC      8#20    8/21  226/17  227/18  229/40  230/38  231/48  238/22
DEGTI  9D1E     44/20  178#11
DFAD10 9D50    178/39  178#43
DFALSE 9D4E    178/13  178/17  178/20  178/21  178/26  178/30  178/34  178#41  179/51
DFCLRS AF75    135/15  251#58
DGETI  9D3C     44/18  178#28
DGTTI  9D2C     44/22  178#19
DHLINE E8BC    235/50  236#19
DIGIT  0511      8#56  186/61  187/13  187/21  187/27
DIN    9753     99/37   99/46  103/19  103/21  103/23  103/25  103/43  104/32  141#13
DINDEX 0057      2#47  153/54
DIO005 9763    131/43  141#28
DIO010 974E    140#50  141/32  141/38
DIV2T9 B330    236/ 7  236/24  269#32
DIVERR 0084      3#43  174/59
DLEVES 00A0      3#52  118/54
DLETI  9043     44/19  178#32
DLIOFF B6D0    135/41  286#29
DLNCTI 9057     44/35  179#49
DLOADA 90A2     83/14  164/46  183#23  244/17
DLTTI  9035     44/21  178#24
DMACT  022F      2#50   89/30   89/38   89/41  222/43  222/45  222/50  222/52
DMACTL D400      4#57  222/46
DMASAV 05E0     10#37   14/ 8   89/27   89/32   89/39
DMASK  05B1      9#60  256/30  253/43  263/39  263/47  268/ 7
DMASKT B3A6    256/24  269#30
DMDGI  9CE5     44/23  176#19
DMOVI  9A85     17/58   24/37   50/12   83/54  116/32  119/20  120/60  121/23  121/56  122/ 6  123/16  123/44  127/13
               127/15  127/40  128/16  148/50  151/37  152/19  152/25  153/21  159#58  165/ 7  165/10  165/17  165/33
               165/37  165/48  166/45  166/52  167/ 6  167/14  167/18  167/29  183/24  183/41  193/20  226/ 9  227/28
               229/41  229/46  229/56  229/58  230/ 9  230/17  230/19  230/27  238/17  238/24  272/38  273/14
[illegible]    05B7      9#40   81/44   81/57   82/14   83/49   83/57   84/12
[illegible]    [illegible]     81/45   81/38   82/18   83#41
DMULI  9C5A     44/16  121/ 6  174# 7
[illegible]    9A06    109/ 9  118#45
```

```
  DNEGI  9CF1     44/33  175/17  175/24  175/49  175/56  176#39  177/ 7  185/38  186/47  206/52  207/39  231/35  231/49
                 232/57  233/ 5  233/ 8  238/59  245/15  249/22  250/11  272/43  273/19
  DNETI  9D25     44/17  178#15
  DNOTI  9D95     44/34  161#42
  DNSIZE 000F      4#40    9/ 7  138/61  144/58
  DOP005 9728     26/12  140# 9  141/36
  DOP010 974E    139/59  140/19  140#49
  DOPEN  96F4    102/19  103/13  104/29  109/17  110/39  137/41  139#28  147/12
  DORI   9D77     44/25  160#56
  DOS    0000      1#14   12/ 6   12/26   34/46   42/40   63/ 5
  DOSINI 000C      2#29   12/16   12/18   12/22   12/24
  DOSVEC 000A      2#51   12/38   12/40
  DOUT   9758    100/33  100/38  102/26  102/28  102/30  102/32  102/42  141#18
  DOWN   0001      5#43  254/11
  DP     00C2      7#39   22/26   49/54   49/56   49/58   49/60   50/11   50/13   52/10   73/49   73/50   73/52   73/54
                  78/19   78/21   78/23   78/25   79/18   91/26  144/49  144/51  144/54  146/39  146/40  147/18  147/20
                 147/22  147/24  151/35  151/43  151/47  151/48  153/10  153/11  153/30  159/28  202/48
  DRAW   0009      4#17  206/48  206/58
  DRAWTO 000A      4#18  205/49  224/34  227/41
  DRKMSK 004E      2#32  279/11  279/15
  DSCMI  9C22    172#22  178/19  178/24  178/28  178/32  230/41  231/55  236/59  237/15  237/36  237/45
  DSISAV 05DC     10#35   12/17   12/19   12/43
  DSPFLG 02FE      2#39   15/38   20/15   81/32   83/23  193/58  194/ 9
  DSTORA 9DA7    165/23  183#40
  DSUBA  9DB1     83/16  164#19
  DSUBI  9C42     44/14  119/16  123/34  123/38  165/20  165/40  165/44  165/52  166/55  166/59  167/21  173#35  175/42
                 184/20  187/ 9  230/11  230/29  230/44  231/58  238/19  238/26  244/32  249/47
  DSUBIX 9C43    173#37  251/22
n DSVSAV 05DE     10#36
  DTAB   0080      7#10   17/18   17/40   17/52   17/57   18/45   18/46   20/55   21/ 5   21/38   22/26   22/27   24/35
                  24/36   24/43   24/59   25/13   33/15   50/10   50/11   52/ 9   52/10   52/21   55/14   55/17   56/21
                  56/23   56/26   66/52   69/17   78/30   79/14   82/34   82/49   82/56   83/ 5   83/13   83/15   83/43
                  83/52   83/53   84/32   84/44   87/18   88/38   88/54   91/26   91/27   94/43  106/14  114/42  114/46
                 114/56  114/57  114/61  115/ 7  116/30  116/31  116/37  116/41  118/28  118/32  118/51  118/57  119/14
                 119/15  119/18  120/30  120/34  120/38  120/42  120/58  120/59  121/ 5  121/ 7  121/21  121/22  121/25
                 121/37  121/38  121/43  121/47  121/48  121/54  121/55  121/61  122/ 5  122/21  122/22  122/25  122/27
                 122/42  122/43  122/53  123/14  123/15  123/25  123/26  123/32  123/33  123/36  123/42  123/43  123/58
                 123/59  124/ 7  124/ 8  124/33  124/37  125/56  126/56  126/57  127/ 8  127/11  127/12  127/14  127/23
                 127/24  127/29  127/32  127/38  127/39  128/10  128/14  128/15  128/18  128/25  128/53  128/54  128/59
                 129/29  129/31  130/17  130/19  132/18  132/21  132/23  132/51  142/41  142/43  142/45  142/47  148/48
                 148/49  148/55  148/60  151/35  151/36  152/17  152/18  152/23  152/24  152/60  153/19  153/20  153/26
                 153/30  155/39  155/40  155/43  155/44  155/48  156/15  156/20  158/37  158/41  159/27  159/28  159/31
                 159/32  159/50  159/51  159/53  159/54  159/58  159/59  159/61  160/ 5  161/14  161/16  161/18  161/20
                 161/43  161/44  162/ 8  162/10  162/48  162/50  162/56  162/61  164/45  164/48  164/51  164/55  164/56
                 165/ 5  165/ 6  165/ 9  165/12  165/15  165/16  165/19  165/22  165/31  165/32  165/35  165/36  165/39
                 165/42  165/43  165/46  165/47  165/50  165/51  166/36  166/37  166/43  166/44  166/47  166/50  166/51
                 166/54  166/57  166/58  166/61  167/ 5  167/12  167/13  167/16  167/17  167/20  167/23  167/24  167/27
                 167/28  167/31  167/32  171/31  171/52  171/53  171/58  171/59  172/22  172/25  173/ 9  173/10  173/11
                 173/13  173/14  173/15  173/37  173/38  173/39  173/41  173/42  173/43  173/45  174/14  174/15  174/20
                 174/23  174/34  174/36  174/55  174/56  175/13  175/20  175/29  175/30  175/38  176/21  176/23  176/41
                 176/42  176/45  176/46  177/ 6  177/33  177/34  177/37  177/46  177/47  177/50  178/43  178/45  180/11
                 180/13  180/34  180/35  180/36  180/37  180/38  180/39  180/56  180/57  180/58  180/59  180/60  180/61
                 181/21  181/22  181/23  181/24  181/25  181/26  181/42  181/44  181/45  181/47  183/23  183/40  183/57
                 184/19  184/37  185/24  185/26  185/28  185/37  186/40  186/42  186/46  187/ 6  187/ 7  193/19  193/43
                 198/35  199/42  199/48  199/50  202/39  202/48  204/22  206/51  207/27  207/33  207/38  207/48  207/49
                 210/52  213/27  213/46  220/ 7  223/12  223/37  224/49  225/12  225/18  225/22  225/37  225/41  226/ 8
                 226/13  226/14  226/17  226/20  226/24  226/30  226/32  227/ 5  227/ 7  227/11  227/12  227/15  227/18
                 227/22   227/27   227/52   227/56   229/26   229/39   229/40   229/44   229/45   229/49   229/50   229/54   229/55
                 229/57   230/ 7   230/ 8   230/10   230/15   230/16   230/18   230/25   230/26   230/28   230/37   230/38   230/40
                 231/48   231/51   231/52   231/54   232/61   233/ 7   234/13   236/50   236/54   237/28   238/15   238/16   238/18
                 238/22   238/23   238/25   238/58   239/36   239/37   240/16   240/18   240/21   240/25   240/29   241/58   242/ 6
                 244/16   244/26   245/14   247/11   247/13   247/14   247/16   247/17   247/19   247/47   247/49   249/ 5   249/ 6
                 249/16   249/31   249/54   250/13   251/11   251/19   262/14   262/15   262/51   272/36   272/37   273/12   273/13
  DTRUE   9D4A   178/12   178/16   178/22   178/25   178/29   178/33   178#38   179/50
  DTX010  9D65   180/12   180#17
  DTXF    9D5E   179/49   180#11
 nDTY010  9D65   180#16
  DUMCAL  A639   220/55   223# 6
  DXURI   9D86    44/26   181#21
  EBNC    0004     5#29    41/ 9
  EBOTOM  0002     4#26   226/38   237/40
  EDGPUL  055E     9#33    88/18   213/23   224/45   225/ 8   230/51   231/20   232/12   232/22   232/39   232/50   242/29   242/56
                 242/61   243/50   251/36
  EDGTAB  800D    37/33    40#60    88/11    88/16    88/23
  EDTABX  000E    37#32   213/16
  EFREE   0008     5#30    41/11   213/24   224/46   225/ 9   242/60   251/35
  EHALT   0002     5#28    41/ 7   230/60   231/29   232/23   232/51   242/57
  ELEFT   0008     4#24   225/45   237/19
  ELEVEL  0002     4#32     4/33
  ENDERR  0081     3#22    62/31
  ENDPT   009E     8#30   229/54   230/15   230/40   231/54
  EOL     009B     2#12    18/22    20/44    21/14    58/32    59/23    69/ 8    70/17    99/38    99/47   100/37   104/49   133/49
                 137/38   139/ 7   140/11   140/17   141/34   144/61   191/31   192/59   195/28   280/45   282/57   283/39   283/41
                 283/43   283/45   283/53   283/61   284/10   284/20
  EONMLS  FFFF     5#37   115/25   117/17   124/58   125/38
  EOPEN   968E    15/23   112/61   133/11   137#27
  EOPERR  0081     3#18    20/35    24/47
  EPUTC   E406     1#22    13/ 6    20/26   109/30   135/16   137/33
  ERIGHT  0004     4#25   225/57   237/ 7
  ESTKP   054E     9#25   197/20   197/35   197/45   198/15   198/33   199/11   199/21   199/35   199/59   199/61   242/43   243/15
                 243/23
  ESTKSZ  000E     4#33     7/24   197/36   199/12
  ETOP    0001     4#27   226/51   237/50
  EWRAP   0001     5#27    41/ 5   230/52   231/21   232/13   232/40
  EXC100  7C08    27/47    28/24    28#52
  EXCMND  7BE2    17/23    28#18
  EXEC    0092     7#22    27/35    28/20    28/53    29/31    47/31    49/37    56/60    58/12    61/10    61/30    64/18    64/26
                  66/38    67/24    70/26    73/15    75/ 9    75/56    81/26    89/21    90/45    91/18    91/46    93/ 9    94/22
                  95/ 7    95/10    95/15    96/36    97/17    99/22   101/13   102/14   103/ 8   104/13   106/11   106/39   107/13
                 108/ 8   109/12   110/15   110/31   110/55   111/16   112/13   112/50   114/15   114/53   116/49   118/39   120/49
                 144/44   146/19   199/31   201/19   203/11   204/16   204/57   205/26   205/57   206/11   206/37   207/ 9   207/44
                 208/13   208/51   208/57   209/ 6   209/23   209/47   210/ 7   210/18   210/35   211/17   211/31   211/59   212/20
                 212/33   212/44   212/59   213/20   213/44   214/ 6   214/49   214/55   215/16   216/35   241/37   274/27
  EXECF   0506     8#50    15/46    28/42    28/56    46/26    46/55    47/18
  EXF     9FD4    61/ 9    61/29    75/54    90/38   104/11   106/ 9   106/37   197#19   204/14   205/55   206/ 9   206/35   206/50
                 207/ 7   207/37   207/42   214/54   215/15   216/34
  EXP192  A00A   197/37   197#55   198/47   199/ 9   199/13
  EXP194  A00C   197#57   199/ 6
  EXPAND  965C   113/ 5   133/12   135/52   136#48
  EXPERR  0002     3#32   197/55
  EXPR    A00F    47/ 7    52/47   198#14
  EXPRC   9FD9   197#24   198/44
  EXPSTK  0093     7#24     8/24     8/34     8/35     8/36     8/37     8/38    47/ 9    47/14    52/48    52/49    61/17    61/20
                  61/37    61/40    76/15    90/50    90/57   104/23   106/14   106/18   106/19   106/42   197/40   197/42   198/35
                 199/16  199/18  199/42  204/22  204/26  205/60  206/ 5  206/14  206/16  206/40  206/42  206/51  207/38
                 207/49  214/58  214/61  215/19  215/22  216/38  216/41  242/46  243/ 9  243/11  243/21  246/18  246/32
                 246/35  272/36  272/37  272/41  272/49  272/51  273/12  273/13  273/17
EXPVAL A014      197/24  197/47  198#17  198/24
FALSE  0000        1#15   13/29  160/11  194/44  208/26  276/18
FCOLOR 0596        9#45  210/45  235/23  253/34  253/36  253/52  263/41
FILERR 0096        3#38  147/56
FILL   0011        4#15  206/55
FILLTO 0012        4#16  205/46
FINE   026E        2#52  137/31
FINEFG 0582        9#61   14/ 7   15/22  112/57  137/30
FLDCLR 059C        9#46  233/40  253/51  264/14
FLOOD  AF98      210/51  253#18
FNAME  96DC      138/46  138#51  144/37
FNOIOB 9870      147/ 8  147#47
FO     0080      285#14  285/15  285/18  285/18  285/18
FPLOT  B256      235/24  253/58  255/21  255/46  256/47  258/47  259/21  263#36  263/37
FSUFER 00A4        3#55  267/39
FSTACK 00F4        8# 5  217/54  217/56  217/57  253/20  253/22  262/14  267/44  267/47  267/49  267/51  267/57  268/ 5
                 268/ 7  268/10
GABRTC AC22      234/46  240#45  256/35
GACC   00CE        8#21    8/22  225/51  225/61  226/ 7  226/ 8  226/13  226/29  226/31  226/44  226/58  226/61  227/ 6
                 227/11  227/27  247/39  247/42  248/20  248/22  248/23  248/25  248/26  248/28  248/29  248/31  248/36
                 248/37  248/38  248/39  249/25  249/34  249/35  249/36  249/37  249/54  250/13  250/17  250/20  250/21
                 250/26  250/27  250/45  250/46  250/49  250/50  250/53  250/54  250/57  250/58
GANGLE 05D2       10#26  223/ 6  223/ 8  239/54  239/56
GBACK  A2C6       44/60  209#17
GBK    A1FB       44/61  206#48
GCHAR  B7E6      204/27  204/33  205/33  285#17
GCHNGE A2E2       44/47  209#42
GCL090 A39F      212#11  212/21  212/34
GCLEAR A3A0       44/49   64/40  212#20
GCLRFM A3AD       44/48  212#33
GCOL   05CF       10#24  220/12  220/51  221/11  221/17  221/19  228/11  228/15  234/61  235/ 6  235/14  235/16  235/59
                 236/ 5
GCOMND 8C39       93/20   93#35   94/41   95/12
GDRW   A20F       44/39  206#58
GDRWTO A185       44/38  205#49
GEDGE  A3DB       44/54  213#16
GETC   0007        1#53  103/16  141/14
GETCOM 7600       15/55   24#27
GETLIN 94B1       25/14   66/53  118/58  132#14
GETR   0005        1#55   25/27  132/26
GEXIT  A391       44/50  211#59
GFIL   A206       44/45  206#55
GFILTO A1B1       44/44  205#46
GFULL  A179       44/51  204#57
GGO    A213       44/43  206#61
GGO090 A1E8      206#27  207/10
GGOTO  A189       44/42  205#52
GGT030 A1DF      206#21  212/54
GHM090 A39F      212#10  212/45
GHOME  A387       44/55  135/27  212#44  213/31
GITER  8C73       94#20   95/31
GJUMP  050A        8#52   13/21   93/50   93/52   93/54
GLT    A24C       45/ 5  207#37
GMODE  A13D       44/59  204#14
```

```
GMOVE  A67A    206/25  207/13  223#61  243/13
GNORTH A3CD     44/56  135/28  212#59
GNUMB  0549      9#23   67/18   67/20   67/29   67/31  225/20  225/24  225/26  225/31  225/32  225/38  225/42  246/ 7
               246/ 8  246/ 9  246/31  246/33  246/34  246/36  246/37  246/39  246/44  246/45  246/46  247/12  247/15
               247/18
GO     0005      4#19  206/61  225/ 5  229/34  233/30  234/22  243/ 6
GODOS  7625     12/35   12#43
GOTO   0006      4#20  205/52  212/52  224/26
GPD    A284     45/ 9  208#57
GPE    A28D     45/ 7  209# 6
GPEN   A275     44/46  208# 9
GPINIT AF60     64/41   93/15  251#29
GPRIOR 026F      2#53  222/23
GPU    A2AB     45/ 8  208#51
GRACTL D01D      4#60  222/41  222/55
GRAFP3 D010      4#59  222/56
GRDLI  84E7    135/32  135/34  279# 5
GREAD  AC5E     55/26  210/47  241#37  243/30
GREFT  8C89     45/ 6   95#25
GRFLAG 0514      8#59   15/20   58/45   93/12  102/25  103/20  112/38  132/60  136/ 5  138/10  241/40  242/21
GRFS   0008      5#53   58/16   61/14  138/28  214/32
GPOPR  05D4     10#27  205/54  206/49  207/ 6  212/53  224/25  224/33  224/61  227/40  229/33  233/29  233/35  234/21
               243/ 7
GROW   05D1     10#25  221/38  228/22  234/59  235/11  235/40
GRSS   0004      5#52   17/11   20/21   66/46   86/35   87/58  138/25  214/32
GRTEMP 05D8     10#31  242/ 5  242/10
GSHADE A31A     44/58  210#29
GSMODE 0537      9#14   87/14  102/27  103/22  112/30  133/53  134/16  134/35  135/40  204/36  205/32  217/50  220/59
               235/49  251/39  253/33
GSOPEN 9510     93/17  103/34  112/34  133#38  204/37  205/13  205/37
GSP094 A172    204#47  205/34
GSPLIT A198     44/52  205#26
GTAB   7E76     37/23   38#32
GTABX  0004     31/11   37#22   93/45
GTEMP  00D2      8#22    8/23  247/40  247/57  248/ 8  248/ 9  248/13  248/14  248/15  248/16  249/13  249/14  249/31
               249/38  249/39
GTEMP2 00D6      8#23  247/43  247/48  247/50  247/54  247/55  248/21  248/24  248/27  248/30
GTLNNO 9F8C     82/31  127/22  128/23  193/21  195#15
GTLSLN 9412    116/29  127#61
GTNOTE 8D57     96/30   96/53   97/ 5   97#54
GTR090 A1E6    206#28  207/45
GTRN   A257     44/41  207#42
GTRNTO A1E9     44/40  206#35
GTT090 A1E8    206#29  206/38
GTURTL A3FB     44/57  213#40
GWALL  A353     44/53  211#13
GX     00EC      7#59   55/14   82/49  213/27  220/ 7  226/14  227/22  228/13  228/17  228/31  238/42  241/45  241/47
               241/50  242/33  243/43
GX1    00BE      8#15    8/16  210/52  224/11  224/14  224/16  225/12  225/18  225/37  227/52  227/56  230/26  238/16
               241/49  241/52  241/58  242/ 6  262/51
GX2    00C4      8#17    8/18  224/49  225/22  225/41  229/26  230/28  238/18
GXNEW  00E6      7#57  205/61  206/ 6  206/22  207/33  212/48  212/49  224/ 7  224/ 9  224/12  228/14  228/18  228/33
               238/41  243/44
GY     00EF      7#60   35/17   52/56  226/24  227/12  228/23  228/27  228/32
GY1    00C1      8#16    8/17  230/ 8  238/23
GY2    00C7      8#18    8/19  230/10  238/25
GYNEW  00E9      7#58  206/15  206/17  206/23  207/27  212/50  212/51  228/24  228/28  228/34
```

```
                    ATARI CAMAC Assembler Ver  1.0A  Page 296
                                                   D1:PILOT.

  HALTFG 05D7     10#30   229/30   231/ 6   231/32   232/26   232/54   233/10
  HITEND 05D6     10#29   227/44   229/32   230/50   231/19   232/11   232/38   233/13
  HITALL 05D5     10#28   227/37   229/31   234/18   234/25
  HMASK  E8CC    236/25   286#25
  HPUSD  D000      5# 5   221/28   222/18
  ICAUX1 034A      1#46     1/47   133/59   134/55   139/34
  ICAUX2 034B      1#47   133/60   139/39
  ICBAH  0345      1#41     1/42    25/25   132/24   142/27
  ICBAL  0344      1#40     1/41    25/23   132/22   142/25
  ICBLH  0349      1#45     1/46    25/33   132/33   139/43
  ICBLL  0348      1#44     1/45    25/31    25/38   132/30   132/50   139/42
  ICCOM  0342      1#38     1/39    25/28   102/22   103/17   109/20   132/27   134/ 5   139/48   140/41   141/21
n ICCOMZ 0022      1#49
  ICDNO  0341      1#37     1/38
  ICHID  0340      1#36     1/37   147/49
  ICOMP  99A9    154/39   155/59   156#45
  ICRLH  0347      1#43     1/44
  ICRLL  0346      1#42     1/43
  ICSTA  0343      1#39     1/40
  IFIND  995B    129/34   151/32   152/12   152/51   155#35
  ILENG  9A1F    156/45   158#60
  IMATCH 99EA    156/46   156/59   157/22   157#50
  IMPERR 0002      3#27    61/51    66/29    70/20    81/19    90/22    95/33    97/47    98/ 9   115/14   117/ 6   119/47   126/26
                 144/31   147/ 5   190/39   209/32   210/15   211/ 5   211/50   213/54   215/35   216/54
  INBFSZ 000A      4#43     9/15    33/12   170/17
  INCCOL B243    231/59   232/49   233/51   255/24   255/43   257/25   258/24   259/16   260/ 8   263#18
  INDENT 05CA     10#20   116/59   190/35
  INIT   7828     13# 6    15/27
  INLN   0080      7#12    15/51    15/53    16/36    17/18    17/40    17/42    17/50    18/17    18/21    18/36    20/31    20/42
                  20/50    20/55    20/61    21/ 5    21/13    22/27    22/32    24/35    24/40    24/41    25/11    25/13    25/22
                  25/24    25/39    25/41    46/44    46/60    48/60    49/23    49/27    49/29    50/33    50/41    51/24    51/31
                  51/33    52/21    52/58    56/43    70/16    78/22    78/24    97/35   100/14   138/53   170/ 7   190/10   190/43
                 192/20   192/44   192/58   198/39   198/45   201/33   202/ 6   202/13   214/41
  INLNBF 0538      9#15    32/59   170/14
  INSERR 0089      3#28   166/11
n INTERR 008C      3#33
  INTEST A60D    213/28   220/ 8   224/50   225/13   225/19   225/23   227/53   236#49   241/59
  INVFLG 0266      3# 8    15/39
  IOCB0  0000      1#29     1/30   132/22   132/24   132/27   132/30   132/32   132/33   132/50   137/27
  IOCB1  0010      1#30     1/31
  IOCB2  0020      1#31     1/32   133/ 9   133/44   133/59   133/60   134/ 5   134/ 7   134/55   135/57
  IOCB3  0030      1#32     1/33    19/13    25/23    25/25    25/28    25/31    25/33    25/35    25/38   102/17   102/24   103/11
                 103/18   103/40   104/27   104/31   104/41   109/15   109/22   109/27   110/37   131/42   148/ 9
  IOCB4  0040      1#33     1/34   147/47
  IOCB7  0070      1#34   148/36
  IOCBSZ 0010      1#28     1/30     1/31     1/32     1/33     1/34   148/34   148/36
  IOEOLN 9492    131#35   132/44   135/59   140/22
  IOEGIN 0504      8#48    15/40   140/ 9
  IOERCK 948E    131#30   141/55   142/16
  IOERR  0086      3#24    21/23    21/35   131/36
  IOHAND 9787    131/26   141/54   142/15   143#17
  IOSTAT 00E4      7#55    21/26    21/38   104/34   131/35   134/10   135/54   141/30   146/23
  IOVBAS F400      1#21    13/ 6    20/26   109/30   134/51   135/16   137/33   141/53   142/14   143/18   143/20
  IVDERR 0002      3#25    27/40    28/37    33/23   270/57
  JNKERR 0002      3#23    29/39
  KAPPLE 0003      6#26    17/54   111/19
  KGETC  E424      1#26   141/53
```

```
                    ATARI CAMAC Assembler Ver  1.0A  Page 297
                                                        D1:PILOT.

   KIN    9774      67/ 8  141#51
   KLOAD  0001       6#24  110/19
   KMERGE 0002       6#25  110/58
   KOFF   0000       5#34   40/46  134/49
   KON    0001       5#35   16/56   40/44  135/24
   LC     0020       5#40   46/31   46/46   50/34   53/ 5   56/45  119/ 7
   LDELET 7AE7      18/33   22#53
   LEND   00E0       7#52  114/46  118/32  120/42  126/56  127/23  146/15  146/42  146/58  147/11  223/61  228/ 7
   LETTRS 0551       9#28   15/21  102/31  103/26  112/23  132/61  135/22  138/13
   LFCOL  0585      10# 6   59/11  134/42  137/44
   LFTCOL 05A3       9#52  255/26  255/28  256/39  256/41  257/10  257/13  257/28  257/31  258/13  258/16  259/ 9  259/12
                   261/46  261/48  262/24  262/26  267/26  267/28
   LFTSTG 878F      73/26   74#15
   LINENO 00DC       7#50   18/13   18/15   23/12   23/14   82/34  116/31  127/24  128/ 6  128/ 7  128/25  129/30  129/32
                   193/23  193/27  193/34  193/43  195/17  195/20
   LINLNG 007A       4#36   10/45   25/30   25/32  132/29
   LINSRT 7ACE      18/42   22#24
   LISTER 90B4     109/25  109/44  114/12  114#21
   LITMAT 05DA      10#33   70/14   70/41
   LITPEN 0000       1#12   53/41   55/29
   LLRG   0002       5#25   40/56
   LMARGN 0052       2#43  137/43
   LMED   0001       5#24   40/54
   LMRGTP 8896     134/41  285#57
   LNFIND 945D     127/ 9  128/50  129#29
   LNOERR 008D       3#34  124/20  126/21  129/54
   LOADFC 0532       9# 9   13/55   16/ 5   17/53   19/ 8   19/12   24/27   25/47  110/23  110/34
   LOGCHF 0000       1#13   38/12   44/27  178/52  181/52
   LP     00BA       7#37   78/30   78/35   78/47   78/55   78/58   79/14   83/43   83/53   84/44  128/10  128/14  128/18
                   148/49  148/55  148/60  155/44  189/13  189/15  189/27  189/29
 n LPENH  0234       2#61
 n LPENV  0235       3# 5
   LS     00DE       7#51   94/29   94/31   94/37   94/39   94/43   94/45   94/46   94/57   94/59  114/42  118/28  120/38
                   126/57  127/ 8  147/14  147/17  147/19  147/28
   LSEND  0005      74/15   74#16
   LSMLL  0000       5#23   40/52
   LSTRN  0569       9#02   28/29   28/61
   LSTRNS 9100     114/21  114/23  115#23  118/10  118/12
   LTRTAB 7FF6      37/31   40#50
   LTTABX 000C      37#30  112/ 9
   LVALTO 7A16      18/28   18#53   19/18
   MAKSTG 876D      73/27   73/33   73/39   73#49
   MALLOC 9AC1     123/46  193/23  164#49
   MATCHP 00FE       8#10   13/60   46/35   46/50   55/54   59/51   59/60   64/49   70/30   71/33   72/17   73/18   80/21
   MATCHR 0533       9#10   72/21   72/23   73/29   73/30   73/31   73/36
   MATSTG 8796      73/32   74#10
   MAXCOL 05AC       9#57  134/29  134/31  231/60  232/ 7  232/43  232/45  265/26  265/29
   MAXLL  2706       4#37  115/24  124/57  129/49  129/50
   MBSPLY 0548       9#56  134/33  134/48  230/55  231/15  266/25
   MMC    0004       7#48  136/28  136/30  136/35  136/36  136/49  136/51  136/61  137/10  165/15  165/35  166/50  167/16
                   168/24  168/31  168/47  169/53  169/11  169/14  169/19  169/24  169/31  169/41
   MTCALL 9538     119/29  123/18  148/52  152/21  166#29
   MOF    0008       7#47  136/14  136/16  137/ 5  137/11  165/ 9  165/46  166/61  167/27  168/37  168/42  169/23  169/25
                   169/33  169/35
   MEMX   0202       7#54  119/12  123/14  123/42  148/48  152/17  152/24  153/19  153/40  153/44  161/27  161/43  164/55
                   165/ 6  165/19  165/38  165/50  165/60  166/ 6  166/30  166/33  166/36  166/44  167/ 5  167/17  167/31
   MEXP   0ED4       7#49  123/28  123/33  123/48  123/54  152/57  152/58  152/60  164/48  165/12  165/43  165/51  165/59
```

```
                166/ 5   166/31   166/34   166/47   166/56   167/24   167/32
  MEMHI   02E5    2#34    13/42    13/45   136/59   137/ 8
  MEMLO   02E7    2#33    13/36    13/38
  MESSUT  84FF   14/32    17/17    21/20    21/33    21/45    81/41    81/54    82/11    82/20    82/44    82/61    83/11    86/40
                 87/ 5    87/13    87/23    88/ 6    88/33    88/43   118/49   118/55   119/40   124/19   124/21   124/30   124/32
                124/36   132/40   280#16   281/15
  MESTAB  B536  280/27   280/29   282#10   282/55
  MFDEL   00F2    7#53    70/33    71/ 9    71/41    72/10
  MLE     78C3   12/37    12/39    15#12
  MLLOAD  7909   15#50    18/51   110/25
  MLOOP   7911   15#55    16/31    17/31    18/34
  MLRES   78E4   15#31    21/48    26/10
  MLRES2  78EC   15#35    17/29
  MLTTMP  05AE    9#58   235/36   236/11   236/27
  MNYNMS  936D  114/25   116/20   118/14   120/23   125#29
  MOD360  AB96  206/45   207/52   213/ 8   231/46   233/ 6   238#55   239/ 6
  MODERR  0022    3#54   204/43
  MOVDA   9BCA  137/18   165/25   167/35   169#11
  MOVIA   9BA6  136/41   165/54   167/ 8   168#29
  MP      00C6    7#40    78/45    78/53    78/56    78/59    91/53    91/55    92/16   159/32
  MSEND   000B   74/18    74#19
  MSKTMP  059D    9#47   263/46   263/50
  MSP     00D6    7#46   136/19   136/21   136/54   136/56   165/ 5   165/31   166/43   166/54   167/12   168/36   168/41   169/18
                169/20   169/32   169/35
n MTSIZ   005A  282#55
  MVINLN  9BF7   32/46   170# 5
  NAMBUF  6E7B   10#47    91/52    91/54    92/40    92/42
  NAMLNG  0542    9#16    91/57    92/15    92/18    92/46
  NCHGMS  009E    3#50   119/39
  NCOLRS  0589   10#10    86/43   134/37   134/59   216/42   217/18
  NEWCUR  AB8C  227/61   238#39
  NEWDEL  AB71  224/29   227/32   238#15
  NEWDRW  A81C  227/35   229#26
  NEWLC   05A5    9#53   256/50   256/52   257/11   257/14   258/49   258/52   261/45   261/47
  NEWLIN  9F98   20/28    21/46    59/14    83/20    83/22    84/42    86/21    86/60    87/20    87/40    88/40    89/ 7   104/43
                105/49   105/52   195#28   280/42   280/51
  NEWPC   05A9    9#55   258/50   258/53   259/11   259/14   259/24   259/26   259/35   259/41   259/51   259/54   260/36   260/39
                261/50   261/52
  NG      0000  285#13   285/17   285/17   285/17   285/19   285/19
  NMBFSZ  000A    9#30     9/31
  NMCERR  0025    3#56   217/29
  NMIEN   D40E    5# 9   135/38
  NMOVI   9477  114/44   114/48   116/39   116/43   118/30   118/34   120/32   120/36   120/40   120/44   130#16
  NMSAF   0553    9#31   114/34   114/35   114/36   114/37   118/21   118/22   118/23   118/24   125/33   125/36   126/ 5   126/ 9
                126/11   130/16   130/18
  NOCLR   0020    1#61   133/57   134/56
  NOCONT  0543    9#17    15/25    16/24    18/10    24/48    25/58    62/32    64/50    64/60    75/42    79/11   119/13   121/60
  NOPLOT  05DB   10#34   224/23   225/16   234/28
  NP      00BE    7#38     8/15    23/13    23/15    23/17    23/19    49/28    49/30    49/31    49/40    50/ 7    50/23    51/32
                 51/34    51/35    51/38    52/ 9    73/57    73/59    73/61    74/ 6    91/59    91/60    92/ 6    92/ 9    92/41
                 92/43    92/45    92/47   146/24   146/26   146/28   146/30   153/ 5   153/ 6   153/26   155/35   155/36   155/40
  NRCERR  0083    3#42    31/39    58/19    66/49    87/47    89/13   112/19   113/12   205/18   215/38
  NS      0080    3#16     3/18     3/19     3/22     3/24     3/28     3/29     3/31     3/33     3/34     3/38     3/42     3/43
                  3/44     3/45     3/46     3/47     3/48     3/49     3/50     3/51     3/55    92/30
  NSTERR  0502    3#21    94/12
  NULACC  9F72   14/14    64/44   194#19
  NULL    0001    3#61    49/16    56/18    66/26    68/53    99/19   125/49
```

```
NUM     0002     4# 5   50/46   52/24   54/35   55/ 9   56/ 7   56/29   93/42   95/28   97/58  125/52  199/ 8  272/10
NUMBER  0088     7#35   17/36   17/38   17/52   18/12   18/14   20/18   20/34   21/ 8   21/22   21/31   21/34   49/52
                49/53   49/57   50/18   50/21   50/49   50/51   50/57   50/61   51/10   54/28   54/31   54/32   54/34
                55/ 7   55/ 8   56/10   56/11   56/25   56/28   69/21   69/28   83/52   84/16   84/20   84/30   84/31
                84/32   84/53   84/60   85/ 5   85/13   94/25   94/26   94/36   94/38   96/33   96/56   97/10  125/56
               126/ 8  126/10  185/21  185/22  185/37  185/49  185/50  185/52  185/54  185/58  185/61  186/ 8  186/11
               186/16  186/17  186/18  186/20  198/56  198/58  199/15  199/17  202/39  272/13  272/15
NUMNAM  7AED    22/24   22/53   23#12  129/33
NVAR    0004     4# 6   50/25   50/46   51/11   56/ 7   66/26   68/56   90/13   93/42   95/28   97/56   99/19  199/ 8
NXTCLR  058A    10#11   86/51  212/37  216/17  217/17  217/23  217/24  252/ 7
NXTIOB  988C   147/53  148/12  148#32
NXTLN   0084     7#13   24/36   24/59   25/61   26/ 6   62/19   62/21   64/36   64/38   77/18   77/20   78/57   78/60
OLLERR  000F     3#36  132/39
ONFTAB  7FEC    37/29   40#42
ONOFFX  000A    37#28   89/17  107/ 9  107/61  213/40  270/29  271/18  271/58
OPEN    0003     1#51  133/61  139/47
OPNBUF  0520     9# 7   99/30   99/51  104/19  104/25  133/48  133/50  137/37  137/39  138/57  139/ 6  142/24  142/26
               144/55  145/ 5  146/27  146/29  149/20
OPR     0040     4#10   52/41  197/32
OPTAB   7E34    37/19   37#39
OPTABX  0000    37#18   48/54
OPTKEY  0004     5#47    5/48   16/52   16/60
OREAD   0004     1#58   99/ 9  103/12  104/28  110/38  133/51  137/40
ORIENT  0560     9#35  221/24  221/43  223/29
OVLPER  009C     3#48  124/31
OWRIT   0008     1#59  100/ 9  102/18  109/16  133/51  137/40
PACTL   D302     4#52   15/32  107/17  107/37
PADDL0  0270     2#57   54/ 9
PCDN    0040     5#21   40/38  208/60  216/ 8  271/40
PCOLR0  02C0     2#55  217/60  222/38
PCTAB   7F70    37/25   39#57   86/18   87/27   87/30  218/25  218/34  219/ 7
PCTABX  0006    37#24  215/58
PCTDN   7FE5    40#38   87/30
PCTUP   7FE1    40#36   86/18   87/27
PCUP    0080     5#20   40/36  208/44  208/54  216/ 5  242/54
PEN     0513     8#58   57/14   87/24   87/36  135/21  208/31  208/39  208/40  208/43  208/45  224/40  235/20  242/27
               242/55  243/52  251/30  251/59
PENCOL  0588    10# 9  208/12  208/21  209/22  209/27  210/34  210/41
PENNUM  0587    10# 8  209/53  210/10  214/38  215/29
PILINI  7800    12#13   12/21   12/23  288/19
n PILLOW  7700    11#16
PILVBL  8491    13/49   13/50  277#12
PKYRDC  D20A     4#55  198/55  198/57
PLOT    AA93   227/59  229/37  233/33  233/61  234/32  235#20
PMBASE  D407     4#58  222/25
PMOVE   9A3B    22/28   52/11   91/28  155/41  155/45  159/29  159/33  159#50
PNCLPS  058E    10#12  135/ 6  216/20  216/46  216/47  217/45  218/17
POINT   008E     7#34   50/10   50/17   50/20   50/50   50/52   50/56   51/ 9   52/38   52/40   67/17   67/19   67/30
                67/32   69/22   69/29   82/27   82/29   90/32   90/34   90/41   90/43   90/51   90/58  114/22  114/24
               116/17  116/19  118/11  118/13  120/20  120/22  125/32  125/35  125/43  125/60  126/14  126/29  127/14
               127/32  127/39  128/15  193/19  193/49  193/55  193/61  195/16  195/19  197/39  197/41  276/13
POFFS   E312   262/19  262/21  262/23  262/25  262/27  262/29  267#44
PP      00CE     7#42  121/22  121/37  121/55  121/61  122/15  122/19  122/25  127/12  128/53  151/36  151/39  152/18
               152/23  153/20  155/43  155/48  155/53  155/56  155/57  156/15  156/20  156/50  156/56  156/57  157/ 5
               157/17  157/19  157/52  157/55  157/56  158/ 6  158/ 7  158/10  158/17  158/60  159/ 5  159/31
PRCLMN  A514    86/54   87/ 7  218#17
n PRGEND  8A99   288#17
```

```
 PRINTC 8F1D    105/ 7  105/11  105#17
 PRNTCL A54F     86/14   87/32  218/39  219# 7  219/12
 PRTEMP 05CC     10#22  218/19  218/41  218/49
 PRTEGS 8981     82/48   82/55   83#29   84/10   86/50
 PRTSFD 89F1     84/ 8   84/38   84#53
 PRTSIG 9797     21/ 6  142#40  202/49
 PSETUP 9A2D    154/37  159#27
 PSF    9F22     17/19   20/56  115/ 5  193#19
 PSTOP  7A3A     20#13   47/39   92/32   93/56   95/35  131/37  149/28  174/60  190/48  194/42  197/57  204/45  204/50
                205/19  208/49  209/34  210/16  210/23  211/ 7  211/51  213/35  213/56  234/57  262/55  267/40  270/61
                275/10
 PTEN   9E79    186/54  186/56  187#46  187/47
 PTENL  000A    187/33  187#47
 PTRIGO 027C      2#59   54/56
 PUSHPS B31F    267/13  267/15  267/20  267/22  267/27  267/29  267/34  267/36  267#57
 PUTC   000B      1#54  102/21  109/19  141/19
n PUTR  0009      1#56
 QDIV   AEBE    226/21  227/19  248#61
 QMULT  AE5B    226/18  227/16  247#37
 QNEGA  AF36    249/23  249/29  249/61  250#43
 R2TMP  009B      8#38  121/21  121/38  121/48  122/43  122/60  123/20
 RADDI  AF50    226/25  227/23  251#11
 RAMTOP 006A      2#30   15/16  102/48  103/50
 RBBACK 0081      6#18  272/44
 RBEYES 0001      6#14  271/22
 RBFWD  0080      6#17  272/40  273/36
 RBHORN 0003      6#16  272/19
 RBINIT B350     14/24  270#14
 RBLEFT 0040      6#19  273/20
 RBOFF  0000      6#12  270/51
 RBON   0020      6#13  270/16  270/39
 RBPEN  0002      6#15  271/44
 RBRGHT 0041      6#20  273/16
 RBTCMD 05C7     10#18  270/17  272/48  274/26  275/22
 RBTOFF B374    212/ 6  270#46
 RBTON  05C5     10#16   14/ 5   56/52   56/55  207/15  207/54  270/38  270/50  271/15  271/31  271/55
 RBTPRM 05C8     10#19  272/50  272/52  273/38  274/29  275/24  275/26
 RBTSNS 05C6     10#17  273/40  274/55
 RBVECT 0502      8#46  270/14  270/26  270/46  275/28
 RDV090 9893    148#38  148/58
 RDYMES B52C     14/34   26/ 9   75/22  109/35  119/31  124/12  281#13
 RDYTXT 0001      3#17  281/14
 REMDEV 989E     14/25   62/29  148#54
 RENERR 009B      3#47  124/18  124/29
 RENFND 9442    121/18  121/26  128#50
 RENNMS 9363    120/19  120/21  124#54
 RESET  86FF     67/43   92#36
 RESET2 8C05     91/22   92#40
 REVRDY B203    256/19  256/25  257/24  257/38  257/51  261/ 8  261/20  261/33  266#43
 REXEC  B425    270/18  272/53  274#48
 REYES  B389     45/13  271#15
 RGCOL  05B6     10# 7   59/ 6   59/20   61/21  134/44  137/46
 RGO    B3D7    207/19  272#36
 RGOFIO B3E9    272/42  272#48  273/18  273/21
 RGSTUP AA7A    234/19  235#10  243/53
 RGTCOL 05A7      9#54  255/51  255/53  258/14  258/17  258/32  258/35  259/36  259/42  259/50  259/53  260/10  260/13
                260/37  260/40  261/12  261/15  261/51  261/53  262/20  262/22  267/33  267/35
```

```
              ATARI CAMAC Assembler Ver  1.0A  Page 301
                                                          D1:PILOT.

 RMOPN  6363    45/15  271#55
 RITEIO 879A    73/38   74#21
 RMARGN 0053     2#44  137/45
 RMSGTB 8879   134/43  285#59
 RONOFF 8350    45/12  270#26
 ROWAC  0076     8#27    5/28  230/ 5  230/ 6  230/18  230/20  230/21  230/37  231/ 9
 ROWCRS 0054     2#46   61/44  212/24  230/45  230/46  230/56  230/59  231/13  231/16  231/25  231/26  235/39  235/45
               240/30  253/26  256/31  256/33  262/30  262/41  266/16  266/48  266/49  267/12
 ROWFLG 059E     9#48  254/ 5  256/10  261/42  266/13  266/22  266/32
 ROWINC 0097     8#26    8/27  254/12  255/10  256/32  262/28  266/44  266/47  267/14
 ROWMAX 8876   134/32  285#51
 RPEN   639A    45/14  271#31
 RROSNS 8400    56/57  273#36
 RSEND  0011    74/21   74#22
 RSUBI  4F57   226/15  227/13  251#19
 RTCLOK 0012     2#42  106/22  106/25  234/41  234/43
 RTMP   0099     8#37  120/58  121/54  122/42  123/32  123/43  123/52  123/58  124/37
 RTURN  83F9   207/56  273#12
 RUN    00FF     9#11   15/36   16/ 8   16/30   20/30   20/52   24/30   25/54   62/14   64/21   64/29   65/ 6   75/19
                77/11   79/ 9  140/15  204/60
 RXDFIV 841B   270/40  270/52  271/23  271/45  272/20  273/39  274#26
 S1H    0080     7#30   24/43   54/27   54/30   75/38   75/40   83/15  136/13  136/15  136/34  136/37  136/53  136/55
               158/37  164/45  165/16  165/22  166/51  166/57  253/19  253/21  262/15
 S1L    00AE     7#29   13/37   13/39   25/60   26/ 5   64/35   64/37   75/37   75/39  189/12  189/14
 S2H    00B4     7#32   13/43   13/46   75/12   75/14  136/24  136/28  136/60  137/ 9  158/41
 S2L    00B2     7#31   13/44   13/47   54/26   54/29   75/13   75/15   83/13  136/18  136/20  136/25  136/29  137/ 6
               137/12  164/51  164/56  165/32  165/39  165/42  165/47  166/37  167/13  167/20  167/23  167/28  189/26
               189/28  267/58
 SATTR  9A86    24/53   83/48  148/61  156/16  162# 8
 SAVCOL 05A1     9#51  235/ 5  235/ 7  235/13  235/15  253/29  253/31  255/32  255/34  262/42  262/44
 SAVIT  88BB    66/33   91#42   99/59
 SAVIT2 88C1    91/13   91#46
 SAVMSC 0058     2#48  102/34  102/39  103/36  103/41  236/19  236/22
 SAVROW 05A0     9#50  234/60  235/10  253/27  262/40
 SAVYR  0512     8#57   20/17   20/41   21/11
 SB     0080    34#28   34/38   34/40   34/42   34/44   34/50   34/52   34/54   34/56   34/58   34/60   35/ 5   35/ 7
                35/ 9   35/11   35/14   35/16   35/18   35/20   35/22   35/24   35/26   35/28   35/30   35/32   35/34
                35/36   35/38   35/40   35/42   35/44   35/46   35/48   35/50   35/52   35/54   35/56   35/58   35/60
                36/ 5   36/ 7   36/ 9   36/11   36/13   36/15   36/17   36/19   36/21   36/23   36/25   36/27   36/29
                36/31   36/33   36/35   37/41   37/43   37/45   37/47   37/49   37/51   37/53   37/55   37/57   37/59
                37/61   38/ 6   38/ 8   38/10   38/24   38/26   38/28   38/34   38/36   38/38   38/40   38/42   38/44
                38/46   38/48   38/50   38/52   38/54   38/56   38/58   38/60   39/ 5   39/ 7   39/ 9   39/11   39/13
                39/15   39/17   39/19   39/21   39/23   39/25   39/27   39/29   39/31   39/33   39/35   39/37   39/39
                39/41   39/43   39/45   39/47   39/49   39/51   39/53   39/59   39/61   40/ 6   40/ 8   40/10   40/12
                40/14   40/16   40/18   40/20   40/22   40/24   40/26   40/28   40/30   40/32   40/36   40/38   40/44
                40/46   40/52   40/54   40/56   41/ 5   41/ 7   41/ 9   41/11   41/17   41/19   41/25   88/13  135/48
               240/26
 SBCMAT 7CAB    31/12   31#59   48/55   89/18   93/46  107/10  108/ 5  112/10  112/47  198/19  211/14  213/17  213/41
               215/59  270/30  271/19  271/35  271/59
 SBCTAB 7E20    31/59   31/61   37#17   37/18   37/20   37/22   37/24   37/26   37/28   37/30   37/32   37/34   37/36
 SBDTAB 80AA    37/41   37/43   37/45   37/47   37/49   37/51   37/53   37/55   37/57   37/59   37/61   38/ 6   38/ 8
                38/10   38/24   38/26   38/28   38/34   38/36   38/38   38/40   38/42   38/44   38/46   38/48   38/50
                38/52   38/54   38/56   38/58   38/60   39/ 5   39/ 7   39/ 9   39/11   39/13   39/15   39/17   39/19
                39/21   39/23   39/25   39/27   39/29   39/31   39/33   39/35   39/37   39/39   39/41   39/43   39/45
                39/47   39/49   39/51   39/53   44#11   45/17   52/37   52/39   93/49   93/51  198/28  198/30
 SBRACK 005B     2#17  191/34
 SC     0090   285#15  285/18  285/18  285/18  285/18  285/18  285/18  285/19  285/19
```

```
                ATARI CAMAC Assembler ver  1.0A  Page 302
                                                   D1:PILOT.


      SCEOA  9EC8      49/33   51/37   78/20  190#10  190/12  190/45
      SCNDEV 97FD      99/10  100/10  101/10  146#15
      SCNEOL 9F18      60/ 8   72/29   80/36  192#58  192/60  201/22
      SCNERR 0085       3#44  149/27
      SCNLBL 9ED3      16/33   18/18   27/25   78/11   80/11   80/14   80/38  190#34
      SCNMOD 0007       4#39  251/38
      SCOMP  9955      79/ 5  154#37
      SCRLTE 8028      37/35   41#15
      SCTABX 0010      37#34  112/46
      SCTEMP 05CB      10#21  217/46  217/61
      SDEL2  98F1     152#17  152/54
      SDELET 98EC      22/54  146/45  152#12
      SDLSTL 0230       2#40  135/43  135/45
      SELKEY 0002       5#46    5/48
      SEN090 9A12     158#20  158/39
      SEND   9A13      78/31   83/44  127/19  128/11  128/20  148/57  155/49  158#37
      SETCLR A4F7     135/ 7  209/28  210/11  215/30  217/22  217#45  251/61  252/ 9  252/13  252/17
      SETCR2 A7E9     227/47  228#10  234/54
      SETCUR ABFF     210/53  227/57  229/27  240#16  242/ 7  262/52
      SETSVL 9EAD      49/45   51/42   74/ 8   63/41  146/32  148/54  189#26
      SETVBV E45C       2#24   13/52
      SFIND  98CE      49/48   51/45  146/35  151#32
      SFNAME 97C2     102/ 9  103/ 5  110/28  144#22  146/16
    n SGETC  E414       1#23
      SGLSTP 0545       9#19   15/37   16/29   17/ 7   17/26   61/34  112/16  205/ 6
      SGSTUF AA67     233/20  234#59  242/24
      SHFAMT 05B0       9#59  236/31  253/46  263/40  263/42  264/ 8  264/10
      SIGNON 0017       3#39   14/31
      SINSRT 9905      22/37   49/61   74/10   91/32  147/26  152#51
      SINTAB AD9F     245/ 9  245#25
      SINVAL AD3B     207/25  207/31  244#14
      SIZEP3 000B       4#61  222/10
      SJUMP  050D       8#53   13/22  198/29  198/31  198/37  199/49  199/51  199/55
      SKCTL  D20F       4#49  196/10
      SKPSEP 9F07      61/28   60/29   96/11   99/14  102/12  109/10  125/45  190/34  192#20  192/22  192/25  206/ 8  209/55
                      214/40  215/14
      SKSTAT D20F       4#50  107/44  107/48
      SLB    9F13      26/15   27/26   29/35   32/44   46/27   46/58   47/20   48/24   64/ 9   78/17   81/ 9   93/22   93/35
                       94/ 6  138/45  191/54  192#44  192/46  197/26  198/17
      SLBYTE B6AC     221/32  286#12
      SMOVI  9A50     153/28  153/31  161#14
      SNXTI  9AAA      24/60   79/15   84/45  115/ 8  122/26  127/33  128/19  149/ 8  156/21  162#48
      SOF    A07C     197/49  199#31
      SP     00CA       7#41  155/39  156/53  156/54  157/13  157/16  157/20  157/25  157/50  157/59  157/60  157/61  158/15
                      159/ 8  159/ 9  159/27
      SPACE  9FA2      82/25   88/56  193/30  193/37  193/41  193/52  195#40
      SPACES 9F9D      82/51   82/58   86/15   86/56  105/35  195#37
    n SPARES 011F      10#38
      SPDVEL AA48      59/26  234/15  234#38
      SPEED  055D       9#32   14/ 6   58/34  106/43  234/38
      SPLFT  0010       1#60  205/29  251/32  285/15
      SPLTSC 0552       9#29  102/29  103/24  112/32  133/52  135/29  138/22  204/30  205/10  205/30  251/33
      SPTERR 0021       3#53  204/48
      SPUTC  E416       1#24  134/51  142/14
      SQUOTE 0027       2#18   24/36   24/39  283/ 7  283/ 9  283/57  283/61
      SSFLAG 02FF       3#12  263/36
      SSKCTL 0232       2#41  196/ 9
```

```
n SSPEL  E41A     1#25
  STAB   678F    73/26   73/32   73/38   73/56   73/60   74/ 5   74#13   74/16   74/19   74/22
  STICKP 0278     2#58   54/45
  STKCC  B2EB   257/48  260/24  261/31  267#19
  STKLC  B2F5   256/22  257/49  267#26
  STKOVF B30D   267#39  267/59
  STKRC  B301   256/23  260/25  261/30  267#33
  STKROW B2E1   256/21  257/47  260/23  261/29  267#12
  STMIST 9E9F    23/20   78/27  128/ 8  189#11
  STPNES 009A     3#46   62/42
n STRTGO 0284     2#60
  STRTKY 0001     5#45    5/48   16/17   16/21
  SVAR   0008     4# 7   51/50   66/26   90/19   91/43   92/37   99/19  144/28  202/31
  SYCMND 7BAB    16/44   18/39   27#25
  TAB    007F     2#19   87/ 9   88/30
  TABADR 0090     7#19   30/42   30/46   31/23   31/25   31/60   32/ 5  105/17  105/18  105/26  105/40  105/41  105/44
  TABLEN 0064    43#35   43/36   45#17   45/19
  TADD1  AE44   207/28  207/34  247#10
  TBLBAS 0090     7#20   32/55   33/ 6   33/15   33/20   33/33
n TCOL   0028     4#41
  TELN   008C     7#17   14/17   14/19   58/22   58/29   58/34   58/39   58/40   58/53   58/56   59/16   59/29   59/33
                 66/52   66/55   67/ 5   67/ 9   67/11   67/45   67/52   68/11   68/36   69/ 7   69/17   70/36   70/37
                 70/44   70/46   70/51   70/54   70/60   71/ 5   71/13   71/23   71/37   71/40   72/ 5   72/ 6   72/13
                 91/27   99/28   99/41   99/50  100/27  100/29  118/57  118/60  131/45  131/49  131/50  201/25  201/26
                202/57  202/58
  TEMP   00A1     7#25   24/56   25/20   26/19   29/14   29/19   29/22   29/23   29/26   32/13   32/45   33/29   33/30
                 50/54   51/13   52/ 8   52/12   56/32   56/33   71/29   71/31   71/52   71/54   71/55   71/60   72/20
                 92/11   92/20  131/20  131/21  131/52  131/53  132/14  132/15  132/48  132/49  135/44  135/46  135/47
                135/49  139/31  139/32  140/38  140/39  140/50  140/51  141/23  141/24  141/51  141/52  142/12  142/13
                157/51  157/53  158/14  158/16  159/ 6  159/11  161/15  161/17  161/19  161/21  161/24  161/25  161/29
                161/31  161/32  161/35  161/36  161/38  161/42  161/44  162/ 9  162/11  162/15  162/19  162/20  162/21
                162/26  162/28  162/31  162/49  162/51  162/55  162/59  162/60  172/24  172/27  174/ 8  174/11  174/12
                174/19  174/21  174/22  174/24  174/26  174/29  174/30  174/33  174/35  175/ 6  175/ 7  175/10  175/11
                175/14  175/21  175/28  175/32  175/35  175/36  175/38  175/46  175/51  175/57  176/20  176/22  185/25
                185/27  185/29  185/31  185/40  185/43  185/53  185/55  186/ 7  186/19  186/39  186/55  186/57  186/58
                187/ 7  187/30  187/42  189/53  189/54  239/41  239/43  244/14  244/15  244/20  244/22  244/26  244/31
                244/36  244/60  245/ 5  245/ 8  245/10  245/14  245/17  246/10  246/11  246/13  246/16  246/21  246/25
                246/26  246/27  246/38  249/10  249/17  249/26  249/41  249/58  250/ 6  274/48  275/ 9  275/12
  TEMP2  00A7     7#26   87/15   87/17   87/18   88/35   88/37   88/38   88/51   88/53   88/54  142/40  142/42  142/44
                142/46  142/49  142/52  142/57  156/51  156/61  157/14  157/24  171/29  171/30  171/31  186/41  186/43
                186/46  186/52  187/ 6  187/10  187/18  187/25  187/36  197/27  197/52  199/34  199/56  218/18  218/23
                218/35  218/38  218/45  218/46  236/49  236/50  236/53  236/58  237/13  237/14  237/27  237/34  237/35
                237/44  237/54  239/31  239/33  239/35  239/37  239/39  280/28  280/30  280/34  281/13  281/16
  TEXBUF BC00    10#43   14/16   14/18   58/25   58/33   67/55   68/14   68/39  100/32  119/ 6  123/10  123/51  202/61
                203/ 9
  TEXLNG 00FE     4#35   10/43   99/43  131/46
  TEXP   A0A7    58/10   67/22   70/24   91/16  100/20  201#19
  TEXT   0020     4# 9   52/31  144/25
  THETA  00F2     7#61   55/20   55/21   83/ 5  206/41  206/43  207/48  213/ 6  213/ 7  231/41  231/42  231/44  231/45
                232/61  238/55  238/58  238/60  239/ 8  239/ 9  239/14  239/15  239/17  239/18  239/21  239/25  239/36
                239/42  239/44  239/48  239/50  239/53  239/55  244/16
  TKNCNT 00FE     6# 7   27/50   28/23   43/37
  TKNNUL 00FF     6# 8   27/53
  TKNOFF 056A     9#43   22/31   22/35   24/57   27/46   27/56   28/18  153/42
  TKNTYP 0568     9#41   24/54   27/45   27/55   28/22   25/30   28/60  153/39
  TMULT  ADF6   207/26  207/32  246# 5
  TONES  0090     3#49  124/35
  TONES  844F     97/32  274/50  276#10
  TOUT   9780    142#12  212/27
  TPBUFF 7700     11#17   11/18   11/22  222/12  222/24
  TRACE  0535      9#11   13/56   16/55   16/57   17/ 6   61/33  108/13
  TRACLR 00F8      8# 7  221/50  223/ 7  223/ 9  223/12  223/15  223/21  223/31  223/33  223/37
  TRAILR A128     66/56  202#57
  TRBUFF 7705     11#19  220/21  221/51
  TRCMES 0018      3#40   17/16
  TRCX   88F8    221/26  286#42
  TRCY   88E0    221/45  286#35
  TRONOF A60C    133/ 6  135/51  213/49  222#34
n TROW   0018      4#42
  TRTCOL 05C4     10#15  214/ 9  222/27  222/37
  TRTINI A5E1     14/23  222# 5
  TRTLOC A57A    220#45  277/13
  TRTLUN 054F      9#26  133/ 5  134/50  135/25  213/47  220/45  222/34
  TRTPLC A558    220# 7  228/ 6  234/55
  TRTSNS 0550      9#27   57/ 7  242/20  243/36  243/37  243/38
  TRYPUS 055F      9#34  220/17  221/46  222/ 8
  TSTCOL B293    233/46  255/14  255/39  256/54  259/28  265#12
  TSTMOD 96B8     17/10   20/20   58/15   61/13   66/45   86/34   87/57  103/27  112/53  138#10  149/16  214/21  214/31
  TSTPIX 827C    233/21  233/58  242/ 9  255/19  255/44  256/44  256/59  257/41  258/26  259/18  260/17  261/23  264# 6
  TSTROW B2B8    253/61  256/ 9  261/41  266#12
  TSTRWX B2D2    266/26  266#34
  TUFLAG 05D9     10#32  220/48  222/ 9  235/12  235/17  235/58  236/ 6  239/52  239/57
  TVBUFF 770C     11#18   11/19
  TXML   0002      5#51  138/19  214/22
  TXOPEN 94F4     14/26   17/14   20/24  103/31  112/27  132#59  212/ 7
  TXSL   0001      5#50   17/11   20/21   66/46  103/28  112/54  138/16  149/17
  TXTCOL 0291      3# 6   58/48   61/26
  TXTROW 0290      3# 7   61/45
  UC     00DF      5#39   70/50  170/12  188/50
  UNDERR 000A      3#30   79/19
  UNTAB  7E67     37/21   38#22
  UNTABX 0002     37#20  198/18
  UP     00FF      5#42  255/ 9
  UPDTAB 7FE1     37/27   40#34
  UPDMAX 0008     37#26  271/34
  USESTK 056B      9#44   62/18   62/20   77/19   77/21   82/26   82/28
  USOERR 008B      3#31   77/27
  USRMAX 0092     30/50   43#37
  USROFF 006C     29/ 6   29/11   30/61   43#36   43/37
  USRTAB 0500      8#44   29/13   29/15   30/41   30/43  104/60  104/61
  USTKP  054D      9#24   18/ 7   62/11   62/17   64/48   75/45   77/14   77/24   77/30   82/22  110/42
  USTKSZ 0030      4#31    9/44   77/15
  USVAR  0010      4# 8   51/55   52/ 5   66/26   90/19   91/43   92/37   99/19  144/28  202/28
  VBAR   007C      2#16   71/ 6
  VDSLST 0200      3# 9  135/33  135/35
  VTHITE 000E    220/18  221/54  223/39  286#51
  VTSENS AC98     57/ 6  242#19
  VTSPLT AC97    242#16  242/22
  VTURT  8910    223/30  223/32  286#49  286/51
n VVBLKD 0224      3#11
  WALLCK AC28    233/22  241# 7  243/33
  WALLS  05C0     10#23   88/47   88/48  135/19  135/20  211/21  211/22  211/42  211/43  211/45  211/46  233/16  233/17
                 241/12  241/16  242/38  242/40  242/52  242/53  243/27  243/29
  WALLTB 8037     37/37   41#23
```

```
 wARPST 000B      2#50   12/13   13/ 9   14/26
 wLTMAX 0012     37#36  211/13
 wMASK  AC3E    211/41  211/44  241/11  241/15  241#20
 wSYNC  D40A      5# 8  279/ 7
 xAE024 85C7     67#42   99/60
 xACCK  8664     43/29   69#47
 xACCPT 8543     42/59   66# 9
 xACCX  8657     43/28   69#38
 xAP090 8FF8    109#39  111/11  111/14  111/17
 xAPPND 9038     42/45  111#10
 xAU010 911A    111/13  116/10  116#16
 xAUTO  9113     42/48  116#10
 xC     0561      9#36  134/20  134/22  225/50  225/60  228/12  228/16  236/57  237/12  240/19  240/22
 xCALL  8703     42/55   75#54
 xCASS  8F86     42/57  107# 9
 xCENTR 8816    134/19  134/21  285#37
 xCM300 8EA7     68/59   91#26
 xCMPUT 8B4B     42/60   90#10
 xCOLRS 8A1F     42/50   86#32
 xCOMM  8F06     42/54  104#58
 xCONT  8534     43/26   64#58
 xCSYNC 8F9C     42/58  107#35
 xDELET 916F     42/37  118#10
 xDIR   8EC3     42/53  104#11
 xDONE  8DF7     43/20  101# 9
 xDUMP  86A6     42/38   81# 9
 xEND   84C2     43/ 5   62# 9
 xENVIR 8AA4     42/51   87#55
 xGRAPH 8C17     43/ 8   93# 9   93/25   93/61
 xIN    8D6A     43/18   99# 9
 xIN060 8D54     97#47   99/20
 xIN090 8D56     97#49   99/11   99/16   99/23
 xITVBV E462      2#23  277/14
 xJMP   880F     43/ 7   77/ 9   78# 9
 xJMPM  8878     43/ 6   80# 9
 xJP005 8814     77/25   77/31   78/ 9   78#17   80/33
 xJUMP  0507      8#51   13/20   15/43   15/45   29/24   29/27   29/32
 xLD090 8FF8    109#41  110/11  110/16  110/29  110/32  110/35
 xLETTR 904A     43/24  112# 9
 xLIST  90A7     42/36  114#12
 xL0005 9001     64/24  110#15
 xL0010 900A    110#23  110/59  111/20
 xL0100 9010     64/15  110/10  110#28  110/52  111/10
 xLOAD  8FFC     42/43  110#10
 xMATCH 866C     43/12   70# 9   73/12
 xMATSX 872C     43/ 9   73# 9
 xMATX  8670     43/11   70#12   73/ 9
 xME090 8FF8    109#40  110/53  110/56
 xMERGE 9026     42/44  110#52
 xMUSE  8732     43/10   73#12
 xNE090 87B6     75#24   75/32
 xNEW   87B9     43/15   75#32
 xNEWV  87A0     43/16   64/33   75# 9   75/35
 xOUT   8DC4     43/19  100# 9
 xPALET 8A09     42/52   86#10
 xPOS   847A     43/21   61# 9
 xREM   8474     42/30   60# 6
```

```
XPEN    91FF     42/49  120#19
XRND10  8511     16/27   64#33
XRUN    84F2     42/39   64# 9
XSAVE   8FCB     42/47  109# 9
XSCROL  907F     43/30  112#46
XSETL   A426     43/23  214#19
XSETP   A431     43/22  214#29
XSLOD   8E5E     43/32  103# 5
XSOUND  8CCA     43/13   96# 9
XSPEED  8F77     43/25  106#37
XSSAV   8E08     43/31  102# 9
XSTOP   84ED     43/27   62#40
XSV050  8FF3    109#35  114/17
XTEMP   00AB      7#27   28/59   29/29   30/40   31/10   31/15   58/38   58/43   58/50   59/ 5   59/10   59/40   62/28
                 62/30   64/43   64/45   66/19   66/25   66/41   67/27   67/28   68/52   69/23   69/31   70/35   71/28
                 71/35   71/36   71/53   72/25   73/21   73/42   78/28   79/ 8   81/25   81/29   81/36   81/49   82/ 5
                 82/ 6   82/33   82/36   83/24   86/38   87/42   87/61   88/ 9   88/21   89/ 8   90/27   90/48   90/52
                 90/60   91/21   91/34   91/42   92/36   93/16   93/18   96/35   96/39   97/31   97/33   97/54   97/61
                 99/13   99/18   99/25   99/26  100/13  100/25  100/26  100/40  102/38  102/51  103/15  103/53  105/19
                105/22  105/46  105/53  106/15  106/17  112/22  112/36  112/59  113/ 6  115/17  116/44  117/ 9  119/50
                121/19  121/27  124/48  125/29  125/41  125/55  126/16  127/61  128/29  204/19  204/39  204/44  204/49
                205/12  205/14  205/36  205/38  207/18  207/20  207/47  207/58  210/50  210/55  212/ 5  212/ 8  213/48
                213/50  215/12  215/27
XTRACE  8FBA     42/56  107#61
XTV     8821     43/17   89#17
XTYPE   83E1     15/42   15/44   42/27   58#10   60/10
XTYPE2  8466     42/28   59#49
XTYPE3  846E     42/29   59#58
XUSE    87E9     42/61   77# 9
XWAIT   8F54     43/14   97/40  106# 9
XXXX    0548      9#22   20/43   21/12  144/38  144/47  146/49  146/61  147/29
YC      0563      9#37  134/24  134/26  226/43  226/57  228/21  237/33  237/43  240/28
YCENTR  8836    134/23  134/25  285#41
```

```
$$$$                    $    $     $$$   $   $  $$$$$  $$$$$
$   $                   $   $ $   $   $  $  $     $    $
$   $                   $  $   $  $      $ $      $    $
$$$$                    $  $   $  $      $$       $    $$$$
$   $                   $  $$$$$  $      $ $      $    $
$   $    $$    $   $  $   $  $   $  $  $     $    $
$$$$     $$     $$$   $   $   $$$   $   $  $$$$$  $$$$$

     DEST=B.JACKIE   USER=B.JACKIE   QUEUE=LPT   DEVICE=@p1


                        PAGES=307


               PATH=:UDD:B.JACKIE:PILOT.PRN

                  $    $$$$$  $   $  $$$$      $
                $ $ $  $      $$  $  $   $   $ $ $
                 $$$   $      $ $ $  $   $    $$$
                  $    $$$$   $  $$  $   $     $
                 $$$   $      $   $  $   $    $$$
                $ $ $  $      $   $  $   $   $ $ $
                  $    $$$$$  $   $  $$$$      $


                 Atari S/W Development HCD1

                      AOS/VS REV 02.08
                   AOS/VS XLPT REV 01.61
```